ABSTRACT

BOZKURT, GOZDE. Curve and Polygon Evolution Techniques for Image Processing. (Under the direction of Dr. Hamid Krim.)

In this digital era of our world, huge amounts of digital image data are being collected on a daily basis. The collected image data is being stored for subsequent processing and use in a wide variety of applications. For this purpose, it is often important to accurately and precisely extract relevant information out of this data. In computer vision applications, for instance, an important goal is to understand the contents of an image and be able to automatically gain an understanding of a scene, implying an extraction and recognition of an object. This task is, however, greatly complicated by the acquired image data being often noisy, and target objects and background bearing textural variations. As a result, there is a strong demand for reliable and automated image processing algorithms, for image smoothing, textured image segmentation, object extraction, tracking, and recognition. The objective of this thesis is to develop image processing algorithms which are efficient, statistically robust and sufficiently general, in order to account for noise and textural variations in images, and which have the ability to extract and provide compact and useful descriptions of target objects in images, for object recognition and tracking purposes.

The main contribution of the thesis is the development of image processing algorithms, which are based on the theory of curve evolution with connections to information theory and probability theory. These connections form the basis for extracting a compact object description, in the form of a polygonal contour. One contribution is the development of a new class of curve evolution equations designed to preserve prescribed polygonal structures in an image while removing noise. In conjunction with these flows, a local stochastic formulation of a well-studied curve evolution equation, namely *the geometric heat equation*, provides an alternative microscopic as well as macroscopic view, which in turn led to our proposal of vanishing at pre-defined directions. Under these flows, the limiting shape of a curve is a polygon, pre-specified by the form and the parameters of the specific flow. The second contribution of the thesis is the development of a new active contour model which merges the desirable polygonal representation of an object directly to the image segmentation procedure by adapting an information-theoretic measure into an active contour framework with an ultimately unsupervised texture segmentation goal. The polygon-propagating models we develop can capture texture boundaries more reliably than the continuous active contour models because the evolution of an active polygon vertex depends on an overall speed function integrated

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
2002

2. REPORT TYPE

3. DATES COVERED
00-00-2002 to 00-00-2002

4. TITLE AND SUBTITLE
Curve and Polygon Evolution Techniques for Image Processing

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
North Carolina State University,Department of Electrical Engineering,Raleigh,NC,27695

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **155**	19a. NAME OF RESPONSIBLE PERSON

along its two adjacent polygon edges rather than on pointwise measurements along continuous contour points. In this way, higher-order statistics which provide more adapted information than the first and second-order, are captured through both the nature of the information-theoretic criterion we utilize, and the nature of the polygon-evolving ordinary differential equations we propose. A supplementary contribution in this sequel is a new global polygon regularizer algorithm which uses electrostatics principles. The final contribution of the thesis is the development of a simple and efficient boundary-based object tracking algorithm well-adapted to polygonal objects. This is an extension of the second contribution of the thesis, and the key idea here is centered around tracking a relatively few vertices together with their corresponding edges, which in turn yields a bookkeeping simplicity and hence efficiency.

The parsimonious set of features provided by the three methods developed in this thesis are useful for object-based description and recognition tasks, and in addition, may provide a viable solution to a parsimonious, and economical representation of large data sets (e.g. a contour represented by a few landmarks).

CURVE AND POLYGON EVOLUTION TECHNIQUES FOR IMAGE PROCESSING

by

GOZDE BOZKURT

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

ELECTRICAL ENGINEERING

Raleigh

2002

APPROVED BY:

Anthony Yezzi

Chair of Advisory Committee

To Alper, my endless love . . .

Biography

Gözde Bozkurt Ünal completed her high school education at Ataturk Anadolu Lisesi, Ankara, in 1992, and received her Bachelor of Science degree in Electrical and Electronics Engineering with High Honors from Middle East Technical University, Ankara, Turkey, in 1996. She started her graduate studies at Bilkent University, Ankara, in 1996, and worked on digital halftoning problems, new error diffusion techniques using adaptive filtering, and inverse halftoning techniques. She earned her Master of Science degree in Electrical and Electronics Engineering from the Bilkent University, Ankara, Turkey, in 1998.

Since August 1998, she continued her graduate studies in the Electrical and Computer Engineering Department at North Carolina State University, Raleigh, NC, and worked as a research assistant at the Vision, Information and Statistical Signal Theories and Applications group, pursuing her Ph.D. degree. Her research interests include curve evolution theory with connections to information theory and probability theory, application of curve evolution techniques to various image and video processing problems such as image smoothing, image and texture segmentation, object tracking, and computer vision problems such as object recognition.

Acknowledgments

In making this work possible, I would like to express my gratitude to:

Dr. Hamid Krim, for his supervision, guidance, financial support through grants from AFOSR, and ONR-MURI; his mentoring towards unexplored and challenging topics, relaying to his students an appreciation for mathematics, and always having an open door for discussions.

Dr. Anthony Yezzi, for his co-supervision, invaluable support and guidance, his insights and ideas, and valuable discussions.

Dr. Kazufumi Ito of the Mathematics Department, for his encouraging comments, and support, and serving on my committee.

Dr. Wesley E. Snyder, and Dr. Gianluca Lazzi, members of my committee, for being interested in this work, and their comments.

Dr. Jean-Pierre Fouque of the Mathematics Department through his enlightening class on SDEs, and for his help and comments; Dr. Yufang Bao for fruitful discussions and suggestions on this topic.

Group and office mates, Dr. A. Ben Hamza and Oleg Poliannikov, for heated discussions on research and life in general, and humorous moments.

Sandy Bronson, our administrative assistant, for being so nice and kind, and making things smoother in arranging official matters, travels, and paperwork.

My parents, Zerrin and Güner Bozkurt, for their sincere love, enormous support and enthusiasm for my education.

Alper Ünal, my beloved husband, who shared with me the pains and the happiness during the last four years; without his endless love and support, this could not have been possible.

Table of Contents

List of Figures

Chapter 1

Introduction

In this digital era of our world, huge amounts of digital image data are being collected on a daily basis. The collected image data is being stored for subsequent processing and use in a wide variety of applications. For surveillance applications for instance, the amount of remotely sensed imagery collected daily by space-borne or airborne systems is on the order of terra bytes. It is often important to accurately and precisely extract relevant information out of this data for a variety of applications. It is, however, impossible to manually carry out this processing and analysis wholly by human operators. The acquired image data is often noisy, and target objects and background bear significant textural variations. It may also be desirable to track features or objects in an image through time to obtain a dynamic analysis of the scene. In computer vision applications, for instance, an important goal is to understand the contents of an image and be able to automatically gain an understanding of the scene and the surroundings, implying an extraction and recognition of an object. As a result, there is a strong demand for reliable and automated image processing algorithms, for image smoothing, textured image segmentation, object tracking in video sequences, and object extraction and recognition.

The broad objective of this thesis is that of developing image processing algorithms which are efficient, in the sense of ease of computations, fast, statistically robust, in the sense of being resilient to noise, statistically significant and meaningful, in the sense of accounting for textural variations in images, with an ability to extract and provide compact and useful descriptions of target objects in images, for object recognition and tracking purposes.

We next give the motivations for the image processing algorithms we developed, which constitute the core contributions of this thesis, and then describe the organization and summary of the thesis.

1.1 Thesis Motivations and Contributions

Curve evolution techniques for image processing involve a propagation or deformation of a curve via certain partial differential equations (PDEs). Curve evolution techniques are applied to a variety of problems, such as image filtering, smoothing, image segmentation, object tracking, shape analysis, and morphological operations. The main contributions of this thesis are, first the development of a new class of curve evolutions as nonlinear filters for curve and image smoothing, second the development of a new class of curve evolutions, where the curve takes the form of a polygon for image segmentation, and which makes use of an information-theoretic measure adapted to texture segmentation. A third contribution of the thesis is the extension of the second algorithm that is developed in Chapter 4 to object tracking in video sequences. Potential applications of the proposed algorithms in shape or object recognition are also alluded to.

The motivation and contribution of each of these algorithms are described in the next three subsections.

1.1.1 A New Class of Curve Evolutions for Nonlinear Filtering

Curve and image evolution techniques have emerged in recent years as important applications of PDEs to nonlinear curve and image filtering. The well-known low-pass filtering of an image by a Gaussian function has been shown by Witkin [147] to be equivalent to evolving an image with a heat equation, which is a linear PDE also known as a linear diffusion. This in turn lead to various developments in nonlinear filtering of images through numerous PDEs and nonlinear diffusion techniques [6, 110, 117, 122, 136]. In addition to these developments in image filtering, the advance of curve evolution techniques may be considered as an implication of the desire to develop shape-based techniques, which is closer in spirit to providing object level knowledge in image analysis. Image evolution equations operate on a pixel-based knowledge, whereas curve evolution equations operate on individual level curves of an image, hence at a higher level than the former. Shape information of an object in an image, may be reminiscent of a curve extracted from an image, thus for an object-oriented filtering, curve evolutions may be more appropriate.

Much of the research in curve evolution theory has centered around the so-called geometric heat equation, which is well-known for its smoothing properties. Indeed, any curve evolved through the geometric heat equation is circularized [54]. One way of overcoming this effect is by using prior information on the preferred directions of important features of objects in an image. Polygonal structures are ubiquitous in images of man-made objects such as buildings, roads and so on, which contain many straight lines, often oriented in particular directions.

The first contribution of this thesis is the development of curve evolution techniques which constitute a new class of nonlinear filters aimed for smoothing along salient lines of an image. These filters which act on level curves preserve sharp corners, and hence preserve the polygonal structures present in an image. This is achieved by a modification, indeed a directional generalization, of the celebrated geometric heat equation. The designed equations or flows preserve certain classes of features in a curve. This, followed a local stochastic formulation of the geometric heat flow, leading to a new macroscopic view of this equation, which is later further specified to vanish at pre-defined directions. The limiting shape resulting from each flow that belongs to the new class of curve evolving flows is a polygon, pre-specified by the form and the parameters of the specific flow. When applied to an image, a selective nonlinear filtering along the salient lines in the image is achieved slowing down the effects of geometric diffusion across important structural directions. This new class of filters are presented in Chapter 3.

We have also developed a variant of this contribution, for smoothing a group of level curves of synthetic aperture radar (SAR) imagery, which is particularly robust to speckle noise typically present in SAR images. These curve evolutions, which lead to a simple segmentation of SAR images and target recognition on the extracted silhouettes, are presented in [140, 142].

1.1.2 A Polygon Evolution Approach to Image and Texture Segmentation

Image segmentation has received a lot of attention through the years, and a vast array of different approaches have been proposed, such as thresholding and local filtering [22, 97, 112, 134], region growing [100], snakes, and active contours [25, 34, 70, 72, 93], and global energy minimizing techniques using various perspectives such as Bayesian, and minimum description length [17, 51, 87, 102]. Following the pioneering snake methodology [70], curve evolution approaches, so-called active contours, have been popular in image segmentation. The key idea in the active contour framework is the construction of an energy functional for the active contour, and its minimization through the gradient descent equations that propagate the active contour. Two main categories of active contour methods are given by; geometric (also called geodesic) active contours [23, 25, 72, 93], where the curve evolution is based on edge information, and region-based active contours [29, 115, 121, 150], where the curve evolution is based on the region information inside and outside the curve.

The region-based active contour models proved to be more robust to noise conditions when compared to edge-based active contour models. The region-based model usually utilizes simple region statistics such as means and variances, hence can not account for higher order nature of the textural characteristics of image regions. In addition, the object delineation by active contour methods, results in a contour representation which still requires a substantial amount of data to be stored for subsequent multimedia applications such as visual information retrieval from databases. Polygonal approximations of the extracted continuous curves are required to reduce the amount of data since polygons are powerful approximators of shapes for use in later recognition stages such as shape matching and shape coding.

The second contribution of this thesis is the development of a new active contour model which nicely ties the desirable polygonal representation of an object directly to the image segmentation process by including an information-theoretic measure into the active contour framework with an unsupervised texture segmentation goal. The polygon-propagating models we develop can robustly capture texture boundaries relative to the continuous active contour models as the evolution of an active polygon vertex depends on an overall speed function integrated along its two adjacent polygon edges rather than pointwise measurements along continuous contour points. In this way, more higher-order statistics than the first and second-order are rationally captured through the proposed information-theoretic measure we utilize, and the nature of the polygon-evolving ordinary differential equations we derive. This new variational texture segmentation model, is unsupervised since no prior knowledge on the textural properties of image regions is used, and will be described in Chapter 4.

A by-product, nevertheless necessary, contribution in this sequel is a new polygon regularizer algorithm which uses electrostatics principles. This is a global regularizer, and is more consistent in preserving local features such as corners than a local polygon regularization, as is explained in Section 4.5.

1.1.3 A Polygon Propagation Approach to Video Object Tracking

Video sequences, i.e. time-varying image sequences, provide additional information on how scenes and objects change over time when compared to still images. The problem of tracking moving objects has a variety of applications in video surveillance, traffic monitoring, video coding, and robotics. Object tracking methods, may be classified into two categories according to the type of information they use. Boundary-based methods, which utilize the boundary information along the object's contour, make use of snake models [12, 67, 70, 89], or geometric active contour models [24, 114], and usually constrain motion by certain motion models such as rigid or affine. Region-based methods [9, 10, 145] segment an image sequence into regions (with different motions), which are matched to estimate motion. The cost of matching regions significantly increases the computational burden of these techniques.

The third contribution of this thesis is the development of a simple and efficient boundary-based tracking algorithm well-adapted to polygonal objects. We build on the insight gained from the the second contribution of the thesis, namely the active polygon framework, to extend it to track polygon vertices in time-varying images. The key idea here is centered around tracking a relatively few vertices together with their corresponding edges, which in turn yield a simplicity and efficiency in bookkeeping. This object tracking method, together with an experimental study of applying active polygons to an object recognition scenario, are presented in Chapter 5.

1.1.4 Connections Among the Contributions

The three main contributions of this thesis may be cast within a unified objective of extracting a compact object description, in the form of a polygonal contour, which leads to an efficient representation of an object crucial to subsequent computer vision applications. The first contribution, in Chapter 3, aims at removing unwanted perturbations on curves while preserving salient features. It drives a curve (a level curve of an image), which is assumed to contain shape information of an object in an image, towards a polygon by straightening the curve out. Inspired by these filters designed for image smoothing, we proceed to a direct polygonal description of an object in an image with a resulting segmentation, which is the second contribution of this thesis, in Chapter 4. This compact description of an extracted object by a handful of vertices, in turn leads to the idea of tracking these features in a time-varying image sequence, as elaborated in Chapter 5. This contribution may also be viewed as a generalization or extension of that of Chapter 4.

The parsimonious set of features provided by the three methods developed in this thesis, are useful for object-based description and recognition tasks, and in addition, may provide a viable solution to a parsimonious, and economical representation of large data sets (e.g. a contour represented by a few points).

1.2 Thesis Summary and Organization

In Chapter 2 we provide a background on the techniques and concepts that are relevant throughout this thesis. A review on image and curve evolution techniques is presented in Section 2.2, and the corresponding numerical implementation technique is given in Section 2.2.3. A brief review of the literature on segmentation methods is presented in Section 2.3. Snakes, and active contour models that are related to the algorithms developed in this thesis are introduced in Section 2.3.3.

Chapter 3 describes a new class of curve evolutions developed for feature-preserving curve and image filtering with a prior knowledge on the salient line directions of objects in an image. The theory of stochastic differential equations (SDEs) is briefly introduced in Section 3.3.1, and a formulation of the celebrated geometric heat equation by a local SDE, and thus obtaining a new microscopic/macroscopic view of this equation, is derived in Section 3.3.2. The insight led to designing a new class of nonlinear filters in the form of diffusions that vanish at pre-defined directions as explained in Section 3.4. These diffusions on curves yield a limiting polygon shape prescribed by the parameters of the flow. We have applied this algorithm to smoothing of structures along known orientation of salient lines in an image while preserving important features. We suggest further applications of the proposed flows in Section 3.5 such as a shape morphing application in computer graphics and a shape recognition scenario.

In Chapter 4, we present a new class of variational active contour models for an unsupervised texture segmentation problem. A brief literature review on texture analysis and segmentation is given in Section 4.2.1. Through a combination of a novel polygon evolution model in Section 4.3, and an information-theoretic criterion adapted as an energy functional for the active polygons in Section 4.4, a robust texture segmentation algorithm is developed as validated through extensive simulation results provided at the end of Chapter 4. A generalization of the proposed active contour model to evolution of multiple active contours is also given. Section 4.5 presents a novel global polygon regularizer idea with a goal of avoiding degeneracy during propagation of the active polygons.

Chapter 5 extends the active polygons acting on a single image to time-varying image sequences, and presents a video object tracking method in Section 5.1, with an application to tracking targets in infra red (IR) image sequences. A brief overview on object tracking methods, and motion estimation methods are also given respectively in Section 5.1.1, and Section 5.1.2. We further demonstrate the utility of the active polygons with an experimental study on an object recognition application in

Section 5.2. This may be considered as part of future research based on the framework set in this thesis.

Conclusions and possible directions of future research are presented in Chapter 6.

Chapter 2

Preliminaries

This chapter focuses on the background of techniques and concepts that are of relevance throughout this thesis.

2.1 Notation

A digital image to be processed is a 2-Dimensional (2-D) function denoted by I, $I : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^2$ is the domain of the function. Processing a function $I^o(x, y)$, which depends on two spatial variables, $x \in \mathbb{R}$, and $y \in \mathbb{R}$, via a partial differential equation (PDE) takes the form;

$$\begin{aligned} I_t &= \mathcal{A}(I, I_x, I_y, I_{xx}, I_{xy}, I_{yy}) \\ I(0, x, y) &= I^o(x, y). \end{aligned} \tag{2.1}$$

Here t is called the *time* or *scale*. I_t denotes the partial derivative with respect to (w.r.t.) t (sometimes shown as $\frac{\partial I}{\partial t}$), and partial derivatives w.r.t. spatial variables are shown inside the operator $\mathcal{A}$ on the right hand side. The solution $I(t, x, y)$ is referred to as a *scale space* for $0 < t < \infty$. If $\mathcal{A}$ is a linear (nonlinear) operator, the scale space is called linear (nonlinear).

Definitions of some operators that are commonly invoked in PDE's often used in computer vision are given next. The *gradient* of $I(t, x, y)$ is a two-dimensional vector (2-D vector) defined as

$$\nabla I \overset{\text{def}}{=} (I_x, \;\; I_y)^T, \tag{2.2}$$

where the superscript T denotes the transpose of a vector. The L_2 norm of the gradient is given by

$$||\nabla I|| \overset{\text{def}}{=} \sqrt{I_x^2 + I_y^2}. \tag{2.3}$$

Vector fields arise when the gradient operator ∇ is applied to a scalar function such as $I(x, y)$. The *divergence* of any vector field $(I(x, y),\ \ J(x, y))^T$ is

$$\nabla \cdot \begin{pmatrix} I \\ J \end{pmatrix} \overset{\text{def}}{=} I_x + J_y. \tag{2.4}$$

The *Laplacian* operator acting on a 2-D function is defined as $\Delta I = I_{xx} + I_{yy}$. It is clear that the Laplacian can also be written as $\Delta I = \nabla \cdot \nabla I$. Throughout the thesis, an inner product (dot product) is denoted by either $< \cdot, \cdot >$, or by a dot $\cdot$. The area of a region R is denoted by $|R|$. A boldface letter is used to denote both a vector and a matrix.

Some background on curve and image evolution techniques, and an overview of image segmentation techniques are presented in the following sections.

2.2 Image and Curve Evolution Techniques

Obtaining a family of images (curves) from an initial image (curve) through a PDE is referred to as an evolution of the image (curve) through time t. It is equivalent to the scale space concept defined in the previous section. We look at some well-known evolution equations for images and curves next.

2.2.1 Image Evolutions

It is known that a low pass Gaussian filter from signal processing may be implemented by evolving the intensities of an image $I^o(x, y)$ via the linear heat equation [147],

$$\begin{aligned} I(0, x, y) &= I^o(x, y), \\ I_t(t, x, y) &= \nabla \cdot \big(\nabla I(t, x, y)\big), \quad t > 0. \end{aligned} \tag{2.5}$$

The solution to this equation yields a parameterized family of new images $I(t, x, y)$, where the image at each time $t > 0$ is equivalent to the original image $I_0(x, y) = I(0, x, y)$ convolved with a

Gaussian filter of variance $2t$. This equivalence gives rise to a natural generalization of the low pass filter using nonlinear diffusion.

Nonlinear diffusion has a distinct advantage in image processing over linear diffusion in that it may be allowed to handle anisotropies (giving rise to the so-called *anisotropic diffusion*) in an image.

A popular approach to anisotropic diffusion is based upon models first introduced by Perona and Malik in [117]. Since then, these models have received a tremendous amount of attention, as have the models based upon curve evolution theory. Perona and Malik extended the linear heat equation by considering diffusion coefficients which vary with the strength of the gradient at different points of an image. This leads to PDE's of the form $I_t = \nabla \cdot \big(g(\|\nabla I\|)\nabla I\big)$, where $g : \mathbb{R} \to \mathbb{R}$ is typically a monotonically decreasing function which suppresses diffusion where the gradient is high (near an edge).

Nonlinear diffusion is particularly important when salient image features are of interest. For example, when the preservation of sharp edges is important, it is natural to consider an anisotropic model which diffuses an image only along the local direction of its edges. One such approach is to consider an image $I(x, y)$ as a collection of iso-intensity contours, or level curves, and to note that at an edge point, the direction of the edge corresponds to the tangent of the iso-intensity contour running through that point. Let η denote the direction normal to the level curve through a given point (the gradient direction), and let ξ denote the tangent direction.

We may write these directions in terms of the first derivatives of the image as

$$\eta = \frac{(I_x, I_y)}{\sqrt{{I_x}^2 + {I_y}^2}}, \quad \xi = \frac{(-I_y, I_x)}{\sqrt{{I_x}^2 + {I_y}^2}},$$

Since these constitute orthogonal directions, we may exploit the rotational invariance of the Laplacian operator and re-write the linear heat equation in terms of these two variables:

$$I_t = \nabla \cdot (\nabla I) = I_{\xi\xi} + I_{\eta\eta}$$

where $I_{\eta\eta}$ and $I_{\xi\xi}$ denote the second-order directional derivatives in the directions of η and ξ respectively. One may then derive the following expressions

$$I_{\eta\eta} = \frac{I_x^2 I_{xx} + 2I_x I_y I_{xy} + I_y^2 I_{yy}}{I_x^2 + I_y^2} \tag{2.6}$$

$$I_{\xi\xi} = \frac{I_y^2 I_{xx} - 2I_x I_y I_{xy} + I_x^2 I_{yy}}{I_x^2 + I_y^2}. \tag{2.7}$$

By subtracting the normal diffusion component (2.6) from the linear heat equation, which diffuses isotropically, we obtain the following anisotropic model, which diffuses along the boundaries of image features but not across them

$$I_t = I_{\xi\xi} = \frac{I_y^2 I_{xx} - 2I_x I_y I_{xy} + I_x^2 I_{yy}}{I_x^2 + I_y^2}. \tag{2.8}$$

We may obtain this same equation in a completely different and much more geometric manner by specifying the evolution of each level curve in the image as seen in the next section.

2.2.2 Curve Evolutions

Let us denote a family of smooth curves $\boldsymbol{C}(p,t) = (\mathcal{X}(p,t), \mathcal{Y}(p,t))$, which is a mapping from $\mathcal{I} \subset \mathbb{R} \times [0,T] \to \mathbb{R}^2$, where $p \in \mathcal{I}$ is a parameter along the curve, and t parameterizes the family of curves. Denote the tangent vector to the curve at p by $\boldsymbol{T} = \boldsymbol{C}' = (\mathcal{X}', \mathcal{Y}')$, and the normal vector to the curve at p by $\boldsymbol{N} = (-\mathcal{Y}', \mathcal{X}')$. We consider *regular* curves whose tangent is never zero ($\boldsymbol{C}'(p) \neq 0$ for all $p \in \mathcal{I}$).

Given $p \in \mathcal{I}$, the *arclength* of a regular parameterized curve $\boldsymbol{C}$ from the point p_o is by definition

$$s(p) = \int_{p_o}^{p} ||\boldsymbol{C}'(p)|| dp, \tag{2.9}$$

where $||\boldsymbol{C}'(p)|| = \sqrt{(\mathcal{X}'(p))^2 + (\mathcal{Y}'(p))^2}$ is the length of the vector $\boldsymbol{C}'$ [41]. Hence $ds/dp = ||\boldsymbol{C}'(p)||$.

The most general deformation of a planar curve $\boldsymbol{C}^o$ is given by

$$\frac{\partial \boldsymbol{C}}{\partial t} = \alpha(p,t)\boldsymbol{T} + \beta(p,t)\boldsymbol{N}, \tag{2.10}$$

$$\boldsymbol{C}(p,0) = \boldsymbol{C}^o(p). \tag{2.11}$$

It can be shown that by a reparameterization of points, the first equation above can be reduced to [75]

$\frac{\partial \boldsymbol{C}(p,t)}{\partial t} = \beta(p,t)\boldsymbol{N}$. Considering curve evolutions which depend only on the curvature function κ of the curve, β can be written as $F(\kappa)$, and a local deformation as a function of curvature may be written as

$$\boldsymbol{C}_t = F(\kappa)\boldsymbol{N} . \tag{2.12}$$

The curvature function is the second derivative of the curve $\boldsymbol{C}$, in the direction of the unit normal $\boldsymbol{N}$. If the curve is parameterized by its arclength parameter s, the second derivative is given by $\boldsymbol{C}_{ss} = \kappa\boldsymbol{N}$. The following flow

$$\boldsymbol{C}_t = \boldsymbol{C}_{ss} = \kappa\boldsymbol{N} , \tag{2.13}$$

referred to as the *Geometric Heat Equation* (GHE), is well known for its smoothing properties. It has been shown by Grayson [54] that any closed, embedded curve evolving according to (2.13) will convexify and smoothly shrink to a single point in finite time, becoming more and more circular along the way. This flow is also referred to as the *curve shortening flow* since it corresponds to the gradient (descent) evolution of the arclength functional. See [74–76] for a more extensive discussion of the many properties associated with this flow. Since the evolution speed is a function of the curvature at each point on a curve, this flow gives rise to a *Euclidean invariant* scale space (see [5, 6, 147]) in which finer features are removed first, followed by coarser features, as the curve evolves. A related flow, based upon the affine geometry of the curve, is given by $\boldsymbol{C}_t = \kappa^{1/3}\boldsymbol{N}$ and shares many of the same properties as the curve shortening flow but gives rise to a more general *affine invariant* scale space (see [5, 124, 125]).

2.2.3 Level Set Method

A new and efficient method for evolving a single iso-intensity contour has recently been proposed, referred to as a level set method [126]. The parameterized curve $\boldsymbol{C}(p,t)$ is embedded into a surface, which is called the level set function $\Phi(x,y,t) : \mathbb{R}^2 \times [0,T] \mapsto \mathbb{R}$. The curve $\boldsymbol{C}$ is the zero-level set of this function $\Phi(x,y,t)$:

$$\boldsymbol{C} = \{(x,y) : \Phi(x,y,t) = 0\}. \tag{2.14}$$

The evolution equation for $\Phi(x,y,t)$ is derived from the constraint that at any time t, we should have $\Phi(\boldsymbol{C}(t),t) = \Phi(\mathcal{X}(t),\mathcal{Y}(t),t) = 0$, and differentiating this constraint with respect to t, we get

$$\begin{aligned} \Phi_t + \Phi_x \mathcal{X}_t + \Phi_y \mathcal{Y}_t &= 0, & (2.15) \\ \Phi_t + (\Phi_x,\ \Phi_y) \cdot (\mathcal{X}_t\ \ \mathcal{Y}_t)^T &= 0, & (2.16) \\ \Phi_t + \boldsymbol{\nabla}\,\Phi \cdot \boldsymbol{C}_t &= 0. & (2.17) \end{aligned}$$

Substituting the general form of the curve evolution equation Eq.(2.12), which depends on local geometry of the curve, into Eq.(2.17) above yields,

$$\Phi_t + \nabla\Phi \cdot F(\kappa)\boldsymbol{N} = 0. \tag{2.18}$$

Noting that the outward unit normal vector can be written as, $\boldsymbol{N} = \nabla\Phi/||\nabla\Phi||$, an evolution equation for Φ is given by

$$\Phi_t = -F(\kappa)||\nabla\Phi||. \tag{2.19}$$

Thus, the curve $\boldsymbol{C}$ evolving according to Eqn (2.12) can be obtained by the zero-level set of the function Φ which evolves according to Eqn (2.19). The selection of the speed function $F(\kappa)$ has been a subject of research [126]. The simplest form where $F(\kappa) = -\kappa$ results in

$$\Phi_t = \kappa\, ||\nabla\Phi||. \tag{2.20}$$

Note that the unit normal and the curvature of a level curve may be expressed as $\boldsymbol{N} = \frac{\nabla\Phi}{||\nabla\Phi||}$ and $\kappa = \nabla \cdot \left(\frac{\nabla\Phi}{||\nabla\Phi||}\right)$. This allows us to rewrite the above equation completely in terms of Φ and its derivatives,

$$\Phi_t = \nabla \cdot \left(\frac{\nabla\Phi}{\|\nabla\Phi\|}\right) \|\nabla\Phi\| = \frac{\Phi_y^2\Phi_{xx} - 2\Phi_x\Phi_y\Phi_{xy} + \Phi_x^2\Phi_{yy}}{\Phi_x^2 + \Phi_y^2} \tag{2.21}$$

giving us a PDE which is identical to (2.8), and is also referred to as the geometric heat equation since it is a result of applying the previous geometric heat equation (2.13) to the zero-level curve of the level set function Φ.

For curve propagation, the level set methods constitute an Eulerian approach in which the underlying coordinate system remains fixed. The parametric formulation of the curve propagation is in a Lagrangian framework in which the coordinate system depends on the parameterization corresponding to the curve's rest position. In Figure 2.1, it can be observed that the level set formulation, where a level curve is embedded into a higher dimensional level set function Φ, may be more advantageous in handling topological changes when compared to parametric evolution methods of curves (also called marker particle methods [126]). This is due to the fact that the zero level set of Φ need not be simply connected, may split, and merge during the evolution.

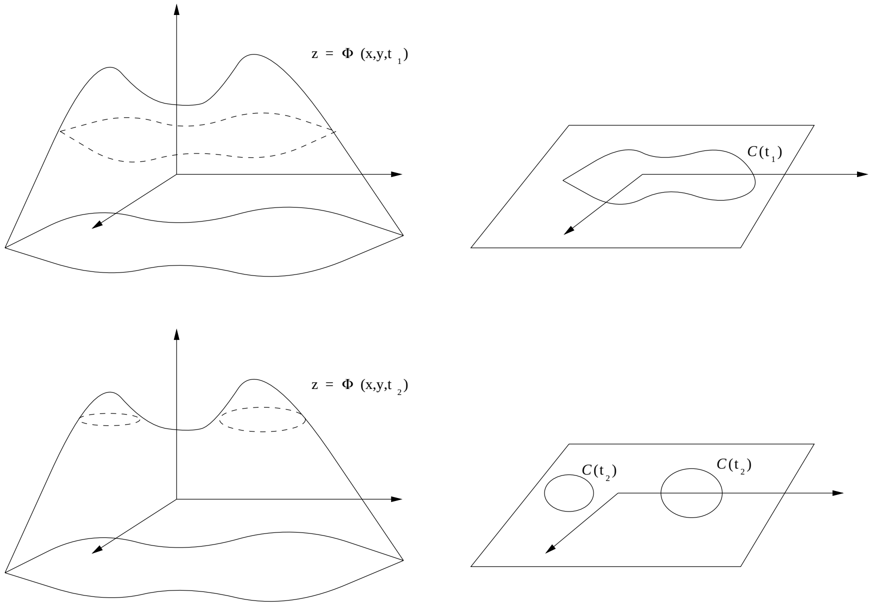


Figure 2.1: Depiction of a level set formulation versus a parametric formulation of a curve. Left column: (Eulerian framework) Evolution of a level set function $\Phi(x, y, t)$ as a graph on a fixed grid is shown at time instants t_1 and t_2. Note that topology changes are handled naturally; Right column: (Lagrangian framework) Evolution of zero level set values of the surface via marker particle methods, i.e. evolution of position vector of a curve, $(\mathcal{X}(p), \mathcal{Y}(p)) = \boldsymbol{C}\,(p)$ is also shown at time instants t_1 and t_2. Notice that it is hard to manipulate topology changes, and serious bookkeeping is required.

The level set function Φ is usually picked as the signed distance function from the zero level curve which is the contour that is to be evolved. A fast level set method so called as narrowband technique [2] developed for propagating interfaces is used in the implementation. To speed up the curve evolution algorithms we develop in this thesis, a periodical re-initialization of the signed distance function, i.e. the level set function, is also carried out by the technique developed by [152].

In the level set equation, (2.19), the speed function F may be given by $F = \alpha + \beta\kappa$, with an advection (first term on the right), and a diffusion term or a curvature term (second term on the right). With a simplest flow when $F = 1$,

$$\Phi_t = -||\nabla\Phi||, \tag{2.22}$$

a curve propagates in its inward normal direction at each point with a unit speed, and it develops singularities, i.e. points that are not differentiable, in finite time [126]. Once such points develop, the normal is not defined at those points, and the propagation becomes ambiguous. Thus, in order to continue the evolution, a *weak solution* is required. Note that a solution is a weak solution of a differential equation if it satisfies an integral formulation of the equation. This in turn implies that the weak solution, as a potential solution, may not require the same degree of differentiability, and allows for more general solutions [126]. A way to obtain such a solution is provided by Sethian, through the notion of an entropy condition, also called Huygen's construction. This is motivated by an analogy of a curve with a propagating flame, and once a point is ignited by the expanding flame (curve), it stays burnt. That is, information once lost, can not be re-created during the evolution. The parallelism of this development and the theory of viscosity solutions of Hamilton-Jacobi equations, as well as shocks and rarefaction fans formations in hyperbolic conservation laws can be found in [126]. To solve the Eq.(2.22), stable numerical schemes are developed, which select the correct weak solution corresponding to the viscous limits of the associated curvature-driven equations. See [86, 88] for detailed discussions on hyperbolic conservation laws. Upwind differencing schemes in computing first order spatial derivatives which use values upwind of the direction of information propagation, are widely employed [126]. The curve evolution equations we develop in this thesis are curvature-driven equations, for which central-differencing schemes for the spatial derivatives and forward-differencing schemes for the time derivatives are adequate. For the implementation of the level set method, and its re-initialization step, however, upwind differencing schemes are used.

2.3 Overview of Segmentation Methods

The problem of image segmentation refers to the partitioning of a domain Ω of a given image I into regions such that each region has properties distinct in some sense. It is expected that the resulting partitions correspond to meaningful parts of objects in an image. Image segmentation is an essential first step in early vision and provides a mechanism for an automatic analysis of image contents. Some important applications include automatic target recognition, remote sensing, automatic visual inspection in manufacturing processes, biomedical image analysis, tracking objects in motion, and so on. In the context of remote sensing of the earth for instance, the image would be partitioned into regions of different terrain or land type.

An initial task for segmentation is to determine the features which delineate and reliably distinguish different regions. Some common features which discriminate the region characteristics are intensity, color, and texture. Following the definition of features for the segmentation problem, it is necessary to select a "good" criterion for capturing and evaluating the features which yield a partition of the image domain into "different" regions.

Existing image segmentation methods may broadly be classified into four groups [153] as described in the next subsections.

2.3.1 Thresholding and Local filtering approaches

Perhaps the earliest approaches to image segmentation are based on thresholding techniques. They rely on a very simple concept which is to compare each pixel value ($I(x, y), (x, y) \in \Omega$) with a parameter (threshold) and decide whether the pixel is within the region or not. The value of the threshold may be globally or locally set [112, 134]. In most methods, the threshold is chosen from the intensity histogram of either the whole image or local regions of the image.

Local filtering approaches for image segmentation are based on detection of edges which correspond to object boundaries or the boundaries between image regions. The celebrated early work by Marr [96] and based on a "primal sketch" concept entailed localizing edges in the image for subsequent use by high level image processing steps. Marr and Hildredth [97] developed an edge detection filter based on local maxima of the gradient magnitude. When the first derivative achieves a maximum, the second derivative is zero. For this reason, an edge detection strategy is to isolate zeros of the second derivatives of I. The differential operator used in these so-called zero-crossing edge detectors, is the Laplacian. In order to mitigate the increase in pixel noise due to differentiation, the image is pre-filtered with a lowpass filter such as a Gaussian kernel. A variant of this idea such as the Canny edge detector [22] uses the zero-crossings of the second order operator in Eq.(2.6) instead of the Laplacian. For efficient implementations, Deriche in [40], derives exact recursive filters, taking a similar analytical approach to Canny.

Segmentation techniques in this group make use of local information, and can often be implemented as a convolution of the given image with the impulse response of a local filter for efficiency sake. They, however, rely on high gradient values in the image to detect prominent boundaries between regions, hence, making them sensitive to noise and affecting the continuity of the edge contours as a result.

2.3.2 Region growing techniques

Region growing techniques partition an image domain into k disjoint regions $O_1, \cdots, O_k$, (i.e., $\Omega = \bigcup_{i=1}^{k} O_i$), such that the image I^o is homogeneous in some sense within each region. Each region O_i is a connected region. Morel and Solimini [100] describe a general multiscale approach to region growing as follows: (i) Initialize the algorithm with the finest possible segmentation at a small scale t, i.e. consider each pixel as a separate region; (ii) Merge all pairs of regions whose "merging" improves the segmentation; (iii) Iterate (go to (ii)) by increasing the scale parameter. Choosing the criterion to perform step (ii) results in different algorithms.

For instance, a piecewise (p-w) constant model for the image is used by [78] as the merging criterion. To a region O_i, the average value of I^o over that region is assigned. Given average values I_i in region O_i and I_j in a neighbor region O_j, the regions O_i and O_j are merged by removing the boundary between them and replacing both I_i and I_j with their weighted average $\frac{|O_i|I_i+|O_j|I_j}{|O_i|+|O_j|}$. Here $|O_i|$ is the area of region O_i. The algorithm looks for a decrease of global energy by merging these regions and by updating the image to I. The global energy is a least squares criterion, i.e. $\int ||I - I^o||^2$ plus a penalty on the total length of the boundaries ℓ. However, the quadratic penalty on the difference between the estimate I and the initial data I^o is not well suited to non-Gaussian noise, such as speckle noise in SAR images.

An earlier method by Pavlidis in [100] also starts by taking all pixels as regions, and merging every pair of regions O_i and O_j such that variance of I^o over $O_i \bigcup O_j$ is less than t (scale).

Variants of region growing or merging methods may yield alternative approaches, e.g. region splitting, and region split-and-merge methods ([100]) which combines the two.

Region growing methods have an advantage of using statistics inside regions, they, however, often generate irregular boundaries [153].

2.3.3 Active contour methods

Active contour models have been widely used in image segmentation applications. The general idea in an active contour framework is to define energy functionals whose local minima comprise a set of solutions, e.g. the boundaries of regions, available to higher level processes.

Snakes

One of the pioneering works in this field is due to Kass, Witkin, and Terzopoulos [70] who addressed the problem of finding salient image contours like boundaries of objects, edges, lines, by so-called "*snakes*" algorithms. They aimed at having the snake lock onto image features by minimizing an integral measure which represents the snake's total energy. By adding suitable energy terms to the minimization, it is possible for a user to push the model out of a local minimum towards the desired solution. Initially, the user places some contour (snake) near an image structure. The constraint forces that act on a snake then push the snake towards features of interest.

Representing the position of the contour parametrically, $\boldsymbol{C}(p) = (\mathcal{X}(p), \mathcal{Y}(p))$, where $p \in [0,1]$, Kass *et.al.* defined the snake's total energy functional, as

$$E(\boldsymbol{C}(p)) = \int_0^1 E_{\text{image}}(\boldsymbol{C}(p))dp + \int_0^1 E_{\text{int}}(\boldsymbol{C}(p)) + E_{\text{con}}(\boldsymbol{C}(p))dp. \tag{2.23}$$

Here, E_{int} represents the internal energy of the snake due to bending:

$$E_{\text{int}} = (w_1(p)||\boldsymbol{C}_p(p)||^2 + w_2(p)||\boldsymbol{C}_{pp}(p)||^2)/2,$$

where w_1 and w_2 control the "tension" and "rigidity" of the snake respectively. (Note that the subscripts denote derivatives with respect to p, and $|| \cdot ||$ denotes the standard Euclidean norm.) Their basic snake model is a spline under the influence of image forces, internal constraints, and other general constraint forces, E_{con}. In an image force, $\int_0^1 E_{\text{image}}(\boldsymbol{C}(p))dp$, $E_{\text{image}}(x,y)$ is a scalar potential field defined on the image plane. The local minima of $\int E_{\text{image}}$ attract the snake. For example, E_{image} can be chosen as an edge functional $E_{\text{edge}} = -||\nabla I(x,y)||^2$, which drives the snake to contours with large image gradients, i.e. the intensity edges.

First Connection between Curve Evolutions and Active Contours

The mathematical foundation of another class of geometric active contours was based on *Euclidean curve shortening*. Defining the length functional (see arclength definition (2.9))

$$L(t) = \int_0^1 \left|\left| \frac{\partial \boldsymbol{C}(t)}{\partial p} \right|\right| dp,$$

then differentiating (taking the “first variation”), and using integration by parts, we have

$$L'(t) = -\int_0^{L(t)} \left\langle \frac{\partial \boldsymbol{C}}{\partial t}, \kappa \boldsymbol{N} \right\rangle ds$$

where κ is the curvature, $\boldsymbol{N}$ is the inward unit normal, and $||\frac{\partial \boldsymbol{C}}{\partial p}||dp = ds$. Thus the direction in which $L(t)$ is decreasing most rapidly is achieved when

$$\frac{\partial \boldsymbol{C}}{\partial t} = \kappa \boldsymbol{N}$$

which defines a *gradient flow*. (For the derivation of this curve length shortening flow, see [148].) A new active contour paradigm was proposed in [25, 72, 148], by changing this ordinary Euclidean arc-length function along a curve $\boldsymbol{C}\,(p)$ with parameter p given by

$$ds = ||\boldsymbol{C}\,_p||dp = (\mathcal{X}_p^2 + \mathcal{Y}_p^2)^{1/2} dp$$

to

$$ds_\phi = \phi ds = (\mathcal{X}_p^2 + \mathcal{Y}_p^2)^{1/2} \phi dp$$

where $\phi(x, y)$ is a positive differentiable function. Thus a new metric is defined with which a new gradient flow is to be found. The gradient flow for the curve shortening relative to the new metric ds_ϕ will be computed, then the first variation of

$$L_\phi(t) = \int_0^1 \left|\left| \frac{\partial \boldsymbol{C}}{\partial p} \right|\right| \phi dp,$$

will lead to

$$\frac{\partial \boldsymbol{C}}{\partial t} = (\phi\kappa - \nabla\phi \cdot \boldsymbol{N}\,)\boldsymbol{N}\,.$$

The metric ds_ϕ has the property that it becomes small where ϕ is small and vice versa. Thus at such points, lengths decrease, and less energy is needed in order to move. If one wants the contour lock onto edges of an image, then it is reasonable to construct a weight which is almost zero near edges, and almost 1 when it is far from the edges. Since $||\nabla I||$ is a local indicator of strength of edges in an image, ϕ is chosen as

$$\phi = \frac{1}{1 + ||\nabla I||}\ .$$

These geometrical snakes are very local models, and are only sensitive to data near the curve.

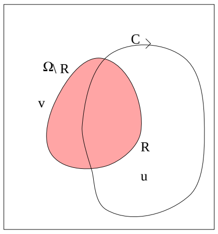


Figure 2.2: A piece-wise constant image consisting of two regions.

The initial contour has hence to be reasonably close to boundaries. The local property also causes these models to be very sensitive to noise.

Region-based Active Contours

Region-based active contours were proposed to overcome the problems with the geometric or geodesic [25, 72, 148] active contours by using both local and global information. The main idea is to assume that the image consists of a finite number of regions, that are characterized by a predetermined set of features or statistics such as means, variances, textures etc. These features/statistics can be estimated from the image data, hence an energy functional may be constructed to pull these statistics apart, i.e. maximize the distance between them in order to separate the corresponding regions. One advantage over the models in the previous subsection is that there is no need to calculate gradients of the image which are usually very sensitive to noise. Region based flows are therefore much more robust to noise, at a cost of additional imposed assumptions on the images, and additional computations.

One such assumption is one's ability to approximate an image by constants, i.e. assuming that the image consists of piecewise constant regions. Let an image consist of only two regions, a foreground R_f, and a background $\Omega\backslash R_f$, and let these be approximated by constants. Fix $\boldsymbol{C}$, an arbitrary closed curve in the image domain Ω, and therefore Ω is partitioned into R and R^c, regions inside and outside $\boldsymbol{C}$ respectively (Figure 2.2). An energy functional to separate the means of the two regions R and R^c, say u and v respectively, is given by Yezzi, Tsai, and Willsky [149]

$$E_{\text{YTW}}(\boldsymbol{C}) = -\frac{1}{2}(u-v)^2. \tag{2.24}$$

A gradient-descent flow on the above energy functional will maximize the Euclidean distance between the mean of region R and the mean of region $\Omega\backslash R$. The energy in (2.24) hence will be minimum when the curve $\boldsymbol{C}$ locks onto the boundary between the foreground (target object), and the background.

A similar energy functional is given by Chan and Vese [29]

$$E_{\mathrm{CV}}(\boldsymbol{C}) = \int\int_R (I-u)^2 dxdy + \int\int_{\Omega\setminus R} (I-v)^2 dxdy. \tag{2.25}$$

The resulting flow in the direction of the gradient descent will automatically move the contour towards the boundary ∂R_f of the target shape, to minimize this fitting energy. For instance, an initial contour encompassing the region R_f will flow inward towards the boundary; while a contour inside R_f will flow outward; and a contour which overlaps R_f will flow in both directions towards the boundary. This makes the initial placement of the contour less restrictive.

The data term, resulting from this formulation, was, however, shown to possibly require additional regularization depending on the particular types of noise, e.g. for salt and pepper type noise, the contour may weave around noisy regions and result in erroneous regions. A regularizing term (penalty on the length of the curve) is added to yield for instance the following energy functional

$$E(\boldsymbol{C}) = E_{\mathrm{YTW}} + \alpha \oint_{\boldsymbol{C}} ds \tag{2.26}$$

where $\oint_{\boldsymbol{C}} ds$ is the total arclength of the curve, and α is a parameter which determines the amount of the desired regularization. The aim is to thus prevent the length of the curve from getting impractically long and producing smooth boundaries. The corresponding gradient descent flow for E_{YTW}, can be shown to be

$$\begin{aligned} \frac{\partial \boldsymbol{C}}{\partial t} &= (u-v)\left(\frac{I-u}{\int_R dxdy} + \frac{I-v}{\int_{\Omega\setminus R} dxdy}\right)\boldsymbol{N} - \alpha\kappa\boldsymbol{N}, \\ &= f_I\boldsymbol{N} - \alpha\kappa\boldsymbol{N}. \end{aligned} \tag{2.27}$$

(A very similar derivation for the first term (data term) is deferred to Chapter 4). The regularizing term (second term above) may also be viewed as a shape prior which is especially very strong at contour points with high curvature. Therefore, there is a trade-off between the data-driven term and the regularizing term. The data-driven term, however, usually dominates the flow.

2.3.4 Global optimization approaches based on energy functionals

The goal of most active contour algorithms is to extract the boundaries of homogeneous regions within an image, whereas the goal of most nonlinear diffusion algorithms is to smooth an image within homogeneous regions but not across the boundaries of such regions. A well-known mathematical model proposed by Mumford and Shah [102] simultaneously addresses both goals. They develop an energy functional which approximates an image by smooth functions in each region instead of constant ones. The Mumford-Shah (M-S) functional is given by

$$\begin{aligned} E(\boldsymbol{C}, f_R, f_{R^c}) &= \int\int_R (f_R - I)^2 dxdy + \int\int_{R^c} (f_{R^c} - I)^2 dxdy \\ &\quad + \int\int_R ||\nabla f_R||^2 dxdy + \int\int_{R^c} ||\nabla f_{R^c}||^2 dxdy + \alpha \oint_{\boldsymbol{C}} ds, \end{aligned} \tag{2.28}$$

where $\boldsymbol{C}$ is the closed, smooth segmenting curve, I is the observed image data, f_R is the smooth function inside the curve, f_{R^c} is the smooth function outside the curve. Minimizing Eq. (2.28) then corresponds to finding estimates f_R and f_{R^c} in regions R and R^c respectively.

The first two terms in the M-S functional are the data-fidelity terms (like the measurement/ observation model), the second two terms are the smoothness terms in the given regions (like a prior model for f given $\boldsymbol{C}$). The last term is a prior model for $\boldsymbol{C}$ which penalizes its arc length. The M-S functional, hence, captures the desired properties of segmentation and reconstruction by piecewise smooth functions as opposed to p-w constant models in Eq.(2.24), and Eq.(2.25).

There are numerous other algorithms based on minimizing different criteria such as Minimum Description Length (MDL) criteria [87], Bayesian criteria [17]. The problem of minimizing energy functionals such as Eq.(2.28) or other energy functionals (MDL or Bayes criteria) is principally computational (e.g. simulated annealing [51]). To overcome this difficulty, algorithms such as mean field annealing [14, 61, 132], graduated non-convexity [17] have also been explored.

Chapter 3

Stochastic Differential Equations and Geometric Flows

In this chapter, we present a new class of curve evolution equations for smoothing of structures along the known orientation of salient lines in curves and images while preserving their important features. The contents of this chapter are outlined as follows. After an introduction in Section 3.1, we briefly review and recap, as was described in Chapter 2.2, some theoretical concepts associated with the curve shortening flow, including its connection to a nonlinear, directional diffusion equation in which image values diffuse locally only along the directions of its edges in Section 3.2. In Section 3.3, we provide a stochastic equivalent equation which in turn unveils a new shape/feature-driven flow described in detail in Section 3.4, which we also believe could offer a variety of applications outside the recognition and classification problems. We conclude with some illustrating and substantiating examples in Section 3.5, and conclusions in Section 3.6.

3.1 Introduction

In recent years curve evolution has emerged as an important application of partial differential equations (PDE's) in image processing, computer vision, and computer graphics. Curve evolution techniques have been applied not only to individual curves, for applications such as edge-detection, skeletonization, and shape analysis, but have also been considered for their simultaneous action on the level sets of an image in a number of geometrically based anisotropic smoothing algorithms. Osher and Sethian [111, 126] extended this latter perspective to the treatment of individual curves through a set of algorithms, known as level set methods, which enable the implementation of curve and surface evolution on a fixed grid. These techniques have aided a number of researchers in pushing the application of curve evolution to new limits by providing a simple framework for treating certain types of singularities such as shocks and topological transitions [109, 111].

Much of the research in curve evolution theory has centered around the so called geometric heat equation [54] in which a curve is evolved along the normal direction in proportion to its signed curvature. This flow is well known for its smoothing properties [74–76] and the fact that it corresponds to the gradient evolution for arclength (thereby earning the name *curve shortening flow*). Because curvature is a purely geometric quantity (invariant to rotation and translation), curvature-based motion gives rise to a Euclidean invariant scale space [5, 6, 147], allowing one to trace features in a curve from finer to coarser scales as the evolution proceeds. An affine invariant scale space can be obtained from a related curvature flow which depends upon the cube root of the curvature (see [5, 124, 125]).

When applied to the level sets of an image, these flows have a powerful denoising effect when run for a short amount of time. If run for too long, however, even large scale features will be destroyed. The reason stems from the fact that as the geometric heat flow shrinks any closed curve, the curve becomes more and more circular (elliptical in the case of the affine flow) and will eventually collapse into a single point [54]. It is therefore not always possible to preserve desired features in the shapes of objects (corners for example) if too much evolution is required to remove a significant level of noise. Furthermore, it is not well understood how these curvature-based filters are affected by different noise distributions and when this sort of problem may occur.

To the best of our knowledge, and aside from [80, 81], nonlinear diffusion in the previous literature was discussed from a purely deterministic perspective. In this chapter, we provide a stochastic formulation of the geometric heat equation and use the resulting insights to develop a new class of curvature-based flows and anisotropic diffusion filters which preserve desired features in the shape of an object. Under these new flows, evolving curves take the limiting form of a polygon (see [20] for evolutions of polygons related to the geometric and affine geometric heat flows, and [144] for evolutions of polygons globally through an electric field concept). The resulting diffusion models may therefore be applied for much longer periods of time without distorting the shapes of polygonal objects in the image, thereby mitigating the tradeoff between noise removal and shape distortion.

Polygonal structures are ubiquitous in images of man-made objects (buildings, roads, vehicles, and so on), which contain many straight lines, often oriented in particular directions (e.g. horizontal and vertical), that come together to form sharp corners. The ability to preserve such distinctive features is not only desirable when filtering an image which contains these types of shapes, but is also important when applying low level smoothing to an extracted shape since such features constitute important and powerful cues for recognizing objects in higher level vision algorithms. We will present both applications in this chapter. From a dual perspective to our contour-based approach to shape representation, skeletonization approaches may also allow shape analysis without displacement of corners [18, 35, 75, 107, 118, 129].

In this chapter, we develop a new class of curve evolutions, which are obtained by a modification of the geometric heat equation. Given an initial shape in the form of a continuous curve, the class of curve evolution equations we will obtain, deform it into a pre-specified final polygonal shape. The problem of deforming an input shape into a different form has been of interest in various fields such as computer graphics [52].

3.2 Background

The geometric heat equation, introduced in Chapter 2, may be obtained as can be recalled in a geometric manner by specifying the evolution of each level curve in the image. Let $\boldsymbol{C}$ denote a particular iso-intensity contour which we will deform over time via the following flow,

$$\boldsymbol{C}_t = \boldsymbol{C}_{ss} = \kappa \boldsymbol{N} \tag{3.1}$$

where s denotes the arclength parameter, κ the Euclidean curvature, and $\boldsymbol{N}$ the inward unit normal. Equation (3.1), referred to as the *Geometric Heat Equation* (GHE), is well known for its smoothing properties. It has been shown by Grayson [54] that any closed, embedded curve evolving according to (3.1) will convexify and smoothly shrink to a single point in finite time, becoming more and more circular along the way. If we apply the geometric heat flow to every single level curve in the image we obtain the same anisotropic diffusion equation that we derived in Chapter 2. To see this, note that at time t each level curve $\boldsymbol{C}^k$ (where the index k distinguishes one level curve from another) is implicitly described by $u(t, x, y) = u^k$ where u^k denotes a particular intensity in the image. Let us choose a parameterization of $\boldsymbol{C}^k$ so that $\boldsymbol{C}^k(t, p) = (\mathcal{X}(t, p), \mathcal{Y}(t, p))$ for $p \in [0, 1]$ and for all $t \geq 0$. We may then write $u\big(t, \boldsymbol{C}^k(t, p)\big) = u\big(t, \mathcal{X}(t, p), \mathcal{Y}(t, p)\big) = u^k$. Differentiating this expression with respect to t yields

$$u_t + \nabla u \cdot \boldsymbol{C}_t = u_t + \nabla u \cdot (\kappa \boldsymbol{N}) = 0.$$

Note that the inward unit normal and the curvature of each level curve can be expressed as $\boldsymbol{N} = -\frac{\nabla u}{\|\nabla u\|}$ and $\kappa = \nabla \cdot \left(\frac{\nabla u}{\|\nabla u\|}\right)$. This allows us to rewrite the above equation completely in terms of u and its derivatives,

$$u_t = \nabla \cdot \left(\frac{\nabla u}{\|\nabla u\|}\right) \|\nabla u\| = \frac{u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy}}{u_x^2 + u_y^2} \tag{3.2}$$

giving us a PDE which is identical to (2.8).

Equation (3.2) is also referred to as the geometric heat equation since it comes from applying the previous geometric heat equation (3.1) to each level curve of an image u. This double meaning of the term *geometric heat equation* is disambiguated by the context in which the flow is applied (i.e. either to an image or to a curve). In this chapter, we will be interested in both cases and will present directional generalizations of the geometric heat flow which are designed to preserve certain types of features either in a curve or in an image. We first however, reformulate the geometric heat flow from a stochastic point of view, giving new insights into the nature and behavior of this nonlinear diffusion model. It was precisely these insights that led us to the generalizations presented in Section 3.4.

Remark: We note that, in general, the popular anisotropic diffusion models of Perona and Malik in [117], are not related to curve evolution theory and are only intended for images, not curves (unless the curve has the form of a graph). As such, we will not attempt to relate the curve evolution models developed in this thesis to Perona-Malik models which represent a different perspective on the subject of nonlinear diffusion.

3.3 Stochastic Formulation of a Geometric Heat Equation

3.3.1 Introduction to Ito Diffusion

The diffusion of a particle is usually well modeled by a Stochastic Differential Equation (SDE) which, in turn, represents the underlying microscopic process of an evolution of a pixel or a point. The dynamics of this evolution at a macroscopic level are captured by a PDE, also referred to as a generator (infinitesimal) of the diffusion [80, 81, 108]. Suppose we want to describe the motion of a small particle suspended in a moving liquid, subject to random molecular bombardments. If $\boldsymbol{b}(t, \boldsymbol{x}) \in \mathbb{R}^n$ is the velocity of the fluid at a point $\boldsymbol{x} \in \mathbb{R}^n$ and time $t \in \mathbb{R}^+$, then a widely used mathematical model for the position $\boldsymbol{X}(t)$ of the particle at time t is an SDE of the form

$$d\boldsymbol{X}(t) = \boldsymbol{b}(t, \boldsymbol{X}(t))dt + \boldsymbol{\sigma}(t, \boldsymbol{X}(t))d\boldsymbol{B}(t), \tag{3.3}$$

where $\boldsymbol{X}(t)$ is an n-dimensional stochastic process, $\boldsymbol{\sigma}(t, \boldsymbol{x}) \in \mathbb{R}^{n\times m}$, and $\boldsymbol{B}(t)$ is an m-dimensional Brownian motion. $\boldsymbol{b}(\cdot,\cdot)$ is called the *drift coefficient*, and $\boldsymbol{\sigma}(\cdot,\cdot)$ is called the *diffusion coefficient*. The first term in this equation corresponds to a non-random/deterministic motion, whereas the second term models randomness or noise in the motion.

The solution of such an SDE may be thought of as a mathematical description of the motion of a small particle in a moving fluid, and such stochastic processes are called (Ito) diffusions [108]. For many applications, a second order partial differential operator $\mathcal{A}$ can be associated to an Ito diffusion $\boldsymbol{X}(t)$ given by Eq. (3.3). The basic connection between $\mathcal{A}$ and $\boldsymbol{X}(t)$ is that $\mathcal{A}$ is the generator of the process $\boldsymbol{X}(t)$. If $w(\boldsymbol{x}) \in C_0^2(\mathbb{R}^n)$, (i.e., it is continuous with continuous derivatives up to order 2, and has a compact support), then $\mathcal{A}$ is given in the form

$$\mathcal{A}\, w = \frac{1}{2}\sum_{i,j}(\boldsymbol{\sigma}\,\boldsymbol{\sigma}^T)_{i,j}(\boldsymbol{x})\frac{\partial^2 w}{\partial x_i\, \partial x_j} + \sum_i \boldsymbol{b}_i(\boldsymbol{x})\frac{\partial w}{\partial x_i}. \tag{3.4}$$

In conjunction with this, the so-called Kolmogorov's backward equation [108], gives a probabilistic solution to linear partial differential equations. Kolmogorov's theorem states that given $\boldsymbol{X}(t) = (X^{(1)}(t), X^{(2)}(t))$, where $E^{\boldsymbol{x}}[\cdot]$ is the expectation operator with respect to the probability law of $\boldsymbol{X}(t)$ starting at the point $\boldsymbol{x}$, and defining $\gamma(t, \boldsymbol{x}) = E^{\boldsymbol{x}}[f(\boldsymbol{X}(t))]$, then there exists an operator $\mathcal{A}$ such that

$$\begin{aligned} \frac{\partial \gamma}{\partial t} &= \mathcal{A}\gamma, \quad t > 0,\ \boldsymbol{x} \in \mathbb{R}^2, \\ \gamma(0, \boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^2. \end{aligned}$$

SDEs and stochastic processes, most commonly the Brownian motion, have previously been used in curve and image analysis. Mumford [101] used it to model completion curves of occluded edges, the so-called elastica. By taking the curvature function (of arc length) as a Gaussian process, and the tangent direction on the curve then as a Brownian motion, he derived the probability of the curves that link occluded edges. For a more general situation, e.g. curves in $\mathbb{R}^3$, Mumford used other sorts of stochastic processes such as an Uhlenbeck process to find the elastica. Williams and Jacobs [146], later in their "Stochastic Completion Fields" work, define the same SDE as Mumford's, for a particle's position and the orientation, and through this model of diffusion incorporate the prior assumption that the maximum likelihood path followed by a particle between two positions and directions is a curve of least energy, and solve it by a discrete formulation. Similarly, a Kalman filter which produces estimates of a system as it evolves in time and affected by noise, (which is indeed an SDE written for the system and its observations), was used in [37] for grouping of contour segments. Our use of SDEs is along a different line of thought in that our inspiration starts with a desired effect of a nonlinear filter. Specifically, the theory of SDEs provides us with a microscopical interpretation of the well-studied geometric heat equation, and leads to a new macroscopic description of this equation which in turn is used to develop a new class of curve evolutions or filters.

3.3.2 Stochastic Formulation of the Geometric Heat Equation

Let us denote by $\theta(t, \boldsymbol{x})$ the angle between the outward normal to the curve and the x-axis at each spatial point $\boldsymbol{x} = (x, y)$. The outward unit normal $\boldsymbol{N}$ can then be expressed in terms of the angle θ as $\boldsymbol{N} = (\cos(\theta(t, \boldsymbol{x}), \sin\theta(t, \boldsymbol{x}))$, which is re-written in terms of $u(\cdot)$ as $\boldsymbol{N}(t, \boldsymbol{x}) = (u_x(t, \boldsymbol{x}), u_y(t, \boldsymbol{x}))/\sqrt{u_x(t, \boldsymbol{x})^2 + u_y(t, \boldsymbol{x})^2}$. It follows, $\theta(u_x(t, \boldsymbol{x}), u_y(t, \boldsymbol{x})) = \tan^{-1}(\frac{u_y(t, \boldsymbol{x})}{u_x(t, \boldsymbol{x})})$. Using these equations, and defining an operator $\mathcal{A}_{\text{GHE}}$ of the form

$$\begin{aligned}
\mathcal{A}_{\text{GHE}} u(t, \boldsymbol{x}) \;=\;& \sin^2\theta(u_x(t, \boldsymbol{x}), u_y(t, \boldsymbol{x}))\, u_{xx}(t, \boldsymbol{x}) \\
& - 2\sin\theta(u_x(t, \boldsymbol{x}), u_y(t, \boldsymbol{x}))\, \cos\theta(u_x(t, \boldsymbol{x}), u_y(t, \boldsymbol{x}))\, u_{xy}(t, \boldsymbol{x}) \\
& + \cos^2\theta(u_x(t, \boldsymbol{x}), u_y(t, \boldsymbol{x}))\, u_{yy}(t, \boldsymbol{x}),
\end{aligned} \tag{3.5}$$

the geometric heat equation (3.2) can be re-written as

$$\begin{aligned}
u(0, \boldsymbol{x}) &= u_o(\boldsymbol{x}), \\
u_t(t, \boldsymbol{x}) &= \mathcal{A}_{\text{GHE}} u(t, \boldsymbol{x}),
\end{aligned} \tag{3.6}$$

where $u_o(\boldsymbol{x})$ is the initial level set function.

In light of the foregoing development, a natural question which arises is: *given a PDE which governs a curve shortening flow, can we obtain a corresponding SDE associated with the underlying diffusion?*

The nonlinearity of GHE presents a significant challenge to find a global Ito diffusion which explains the overall microscopical behavior of the system. Our approach here for solving such a nonlinear problem is, to explore the short-time behavior by linearizing around a known (nominal) solution. The perturbation equations so obtained will be linear and hence an approximate solution to the nonlinear problem can be obtained as the nominal value plus the perturbation term. Let us denote by $u^n(t, \boldsymbol{x})$ the solution to Eq (3.6):

$$\frac{\partial u^n}{\partial t} = \sin^2\left(\tan^{-1}(\frac{u^n_y}{u^n_x})\right)\, u^n_{xx} - \sin\left(2\tan^{-1}(\frac{u^n_y}{u^n_x})\right)\, u^n_{xy} + \cos^2\left(\tan^{-1}(\frac{u^n_y}{u^n_x})\right)\, u^n_{yy},$$

and if we write $u(t, \boldsymbol{x})$ as

$$u(t, \boldsymbol{x}) = u^n(t, \boldsymbol{x}) + \epsilon\, u(t, \boldsymbol{x}),$$

and define the corresponding nominal angle $\theta^n(t, \boldsymbol{x}) = \tan^{-1}(\frac{u^n_y(t,\boldsymbol{x})}{u^n_x(t,\boldsymbol{x})})$, we get a linearized version of the geometric heat equation around a nominal value:

$$\begin{aligned} \frac{\partial u(t, \boldsymbol{x})}{\partial t} &\approx \mathcal{A}_{\text{GHElin}} u(t, \boldsymbol{x}) \\ &= \sin^2(\theta^n(\boldsymbol{x}))\, u_{xx}(t, \boldsymbol{x}) - \sin(2\theta^n(\boldsymbol{x}))\, u_{xy}(t, \boldsymbol{x}) + \cos^2(\theta^n(\boldsymbol{x}))\, u_{yy}(t, \boldsymbol{x}) \\ &\quad + c(\boldsymbol{x})(-u^n_y(\boldsymbol{x})\, u_x(t, \boldsymbol{x}) + u^n_x(\boldsymbol{x})\, u_y(t, \boldsymbol{x})), \end{aligned} \tag{3.7}$$

where $c(\boldsymbol{x}) = \frac{1}{(u^n_x(\boldsymbol{x}))^2 + (u^n_y(\boldsymbol{x}))^2}\left[\sin(2\theta^n(\boldsymbol{x}))(u^n_{xx}(\boldsymbol{x}) - u^n_{yy}(\boldsymbol{x})) - \cos(2\theta^n(\boldsymbol{x}))2u^n_{xy}(\boldsymbol{x})\right]$ (see Appendix A for details of this derivation).

In light of this, we can proceed to state the following:

Proposition 1 *The right hand side of the linear PDE in Eq. (3.7) is the generator of the following Ito diffusion satisfying the SDE*

$$\begin{pmatrix} d\boldsymbol{X}^{(1)}(t) \\ d\boldsymbol{X}^{(2)}(t) \end{pmatrix} = c(\boldsymbol{X}(t)) \begin{pmatrix} -u^n_y(\boldsymbol{X}(t)) \\ u^n_x(\boldsymbol{X}(t)) \end{pmatrix} dt + \sqrt{2} \begin{pmatrix} -\sin\theta^n(\boldsymbol{X}(t)) \\ \cos\theta^n(\boldsymbol{X}(t)) \end{pmatrix} dB(t). \tag{3.8}$$

Proof 1 *The operator $\mathcal{A}_{GHElin}$ in Eq. (3.7) is first re-written as,*

$$\begin{aligned}\mathcal{A}_{GHElin} &= \boldsymbol{b}^T(\boldsymbol{X}) \cdot \boldsymbol{\nabla} + \frac{1}{2}\boldsymbol{\sigma}(\boldsymbol{X})\boldsymbol{\sigma}^T(\boldsymbol{X}) \odot \mathbf{H} \\ &= c(\boldsymbol{X}) \begin{pmatrix} -u_y^n(\boldsymbol{X}) \\ u_x^n(\boldsymbol{X}) \end{pmatrix}^T \cdot \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} \\ &+ \begin{pmatrix} \sin^2\theta^n(\boldsymbol{X}) & -\sin\theta^n(\boldsymbol{X})\cos\theta^n(\boldsymbol{X}) \\ -\sin\theta^n(\boldsymbol{X})\cos\theta^n(\boldsymbol{X}) & \cos^2\theta^n(\boldsymbol{X}) \end{pmatrix} \odot \mathbf{H},\end{aligned}$$

where $\mathbf{H}$ is a Hessian operator and $\odot$ is a Hadamard product. The factorization of $\frac{1}{2}\boldsymbol{\sigma}\,\boldsymbol{\sigma}^T$ leads to

$$\boldsymbol{\sigma}(\boldsymbol{X}) = \sqrt{2}\begin{pmatrix} -\sin\theta^n(\boldsymbol{X}) \\ \cos\theta^n(\boldsymbol{X}) \end{pmatrix} \quad \textit{and by identification,} \quad \boldsymbol{b}(\boldsymbol{X}) = c(\boldsymbol{X})\begin{pmatrix} -u_y^n(\boldsymbol{X}) \\ u_x^n(\boldsymbol{X}) \end{pmatrix}.$$

Given the functions $\boldsymbol{b}(\boldsymbol{X}(t))$, and $\boldsymbol{\sigma}(\boldsymbol{X}(t))$, we come up with a pair of processes $(\boldsymbol{X}(t), B(t))$ such that the SDE in Eq. (3.8) holds. In this case, the solution $\boldsymbol{X}(t)$ is called a weak solution, as it does not specify beforehand the explicit representation of the white noise, i.e. the version $B(t)$ of the Brownian motion is not given in advance.

Both the drift and diffusion coefficient vectors of this SDE are in the tangent direction of our level curves, which helps us interpret it as a 1-dimensional Ito diffusion on the instantaneous tangent direction $\boldsymbol{T}(u_x^n(t), u_y^n(t))$. A differentiability assumption on $u(t,\boldsymbol{x})$

$$\lim_{\delta t \to 0} \frac{u(t+\delta t,\boldsymbol{x}) - u(t,\boldsymbol{x})}{\delta t} = \frac{\partial u(t,\boldsymbol{x})}{\partial t} \approx \mathcal{A}_{\text{GHElin}} u(t,\boldsymbol{x})$$

is sufficient for a short-time existence of the linearized PDE version of the nonlinear geometric heat equation.

Using Kolmogorov's theorem cited in Section 3.3.1, and assuming that $u(t,\boldsymbol{x})$ and its derivatives are "sufficiently regular" (Lipschitz properties), starting at each time t, the diffusion $\boldsymbol{X}(t)$ in Eq. (3.8) is constructed for each time interval $(t-\delta t, t)$, and may be used to write a Backward Kolmogorov Equation,

$$u(t-\delta t,\boldsymbol{x}) = E\{u(t,\boldsymbol{X}(t))/\boldsymbol{X}(t-\delta t) = \boldsymbol{x}\},$$

as a mean value around each pixel dictated by the motion of the constructed diffusion process

$\boldsymbol{X}\,(t)$. This equation can also be written in forward time (since in the small time step δt, the approximate constant-coefficient PDE gives rise to a time-homogeneous diffusion $\boldsymbol{X}\,(t)$ with $b(\boldsymbol{X}\,(t))$ and $\boldsymbol{\sigma}\,(\boldsymbol{X}\,(t))$ to give way to an averaging process in the tangent direction of a level curve in the course of a forward evolution (i.e., estimate the new pixel value at time t as a mean value of two neighboring pixel values on the tangent at time ($t-\delta t$) (See Fig. 3.1)).

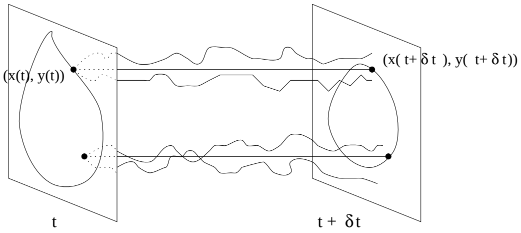


Figure 3.1: Illustration of averaging behaviour. Points of the zero-level set, i.e. initial contour ($\mathcal{X}(t), \mathcal{Y}(t)$), at time t, is shown on the left. Those points whose sample realizations result in an average value of zero at time $t+\delta t$ ($u(t+\delta t, \boldsymbol{x}\,) = E^{\boldsymbol{x}}\,[u(\boldsymbol{X}\,(t))] = 0$) form the new contour($\mathcal{X}(t+\delta t), \mathcal{Y}(t+\delta t)$) (on the right).

This also leads us to infer that locally, we can write a valid diffusion for each time interval $(t-\delta t, t)$,

$$d\boldsymbol{X}\,(t) = \boldsymbol{T}^{\,n}(\boldsymbol{X}\,(t))\;(c^n(\boldsymbol{X}\,(t))\;dt + \sqrt{2}\;dB(t)) \tag{3.9}$$

where $\boldsymbol{T}^{\,n}(\boldsymbol{X}\,(t))$ denotes the known tangent vector at time t, and $c^n(\boldsymbol{X}\,(t))$ is the known drift coefficient at time t, which is $c^n = \frac{1}{\sqrt{(u_x^n)^2+(u_y^n)^2}}\left[\sin(2\theta^n)(u_{xx}^n - u_{yy}^n) - \cos(2\theta^n)2u_{xy}^n\right]$. Apart from a drift on $\boldsymbol{T}$, i.e.,

$$d\boldsymbol{X}\,(t) = \sqrt{2}\,\boldsymbol{T}^{\,n}(\boldsymbol{X}\,(t))\;d\tilde{B}(t), \tag{3.10}$$

the underlying particle motion can be interpreted as a Brownian Motion (BM) on a local frame in the direction of the tangent. Since BM is an averaging process, this SDE explains that the geometric heat equation smooths iso-intensity contours maximally. On a discrete lattice Brownian motion is captured by a random walk with equally likely (i.e., prob. 1/2) displacements to u_T^+ and u_T^-. The latters are obtained by a bilinear interpolation around u (in x and y direction and along the tangent, See Fig. 3.2).

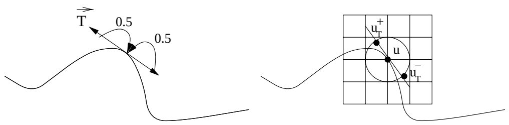


Figure 3.2: Symmetric random walk on the tangent direction, and corresponding interpolation on square grid.

As a result, we can write such an equation as

$$\frac{\partial u}{\partial t} = \frac{1}{2}\, \Delta_T\, u \simeq \frac{u_T^{+} - u}{2} + \frac{u_T^{-} - u}{2}, \tag{3.11}$$

where Δ_T is the Laplacian operator in a tangent direction $\boldsymbol{T}$. Summing up, microscopic dynamics of the system captured by the local diffusion $\boldsymbol{X}\,(t)$ lead to a new macroscopic description of the scenario, i.e., the random walk obtained in terms of the macroscopic variable $u(t, \boldsymbol{x}\,)$.

For simulation purposes, we use a level set methodology [111], which in an Eulerian framework has an advantage of naturally handling topological changes on the level set function. A simulation example where a “T” shape is evolved via a generator of a random walk in a tangent direction, Eq (3.11), is shown in Fig. 3.3.

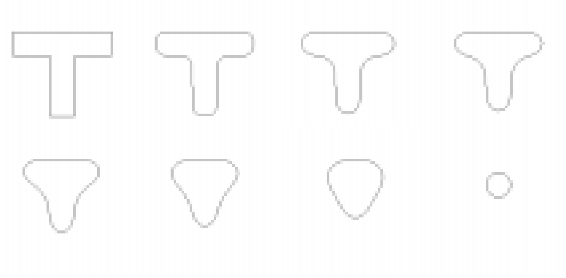


Figure 3.3: Generator of symmetric random walk on the tangent direction implemented on the level set function $u(x, y)$ for a T shape. The tangent direction is estimated directly from the level set set function : $\theta_T = \tan^{-1}\left(\frac{-u_x}{u_y}\right)$ The level set function is on a 250×250 grid, $\delta t = 0.25$.

Practical equivalence of GHE and random walk on the tangent direction is tested by several shapes. Another illustrating simulation is shown in Figure 3.4.

Our neglecting the drift led to an unbiased random-walk on the tangential direction and is validated by the simulation examples presented above, as the generator of symmetric random-walk implementation results are in agreement with the geometric heat equation implementation. Theoretically, a stronger validation is due to Girsanov theorem (see [108]), which says that if we change the drift coefficient of a given Ito process, then the law of this process does not change dramatically, indeed, the trajectories of the process (distribution) change via the measure change on the trajectories. This theorem involving a change of measure provides us with a means of changing the mean of the process $\boldsymbol{X}\,(t)$ we obtained in Eq. (3.9), particularly removing the drift and obtaining the process in Eq. (3.10), where only the version of the Brownian motion changes.

This intuitively appealing interpretation of a particle/pixel motion in the process of a diffusion is shown in the next section to be particularly useful and insightful for developing more general and

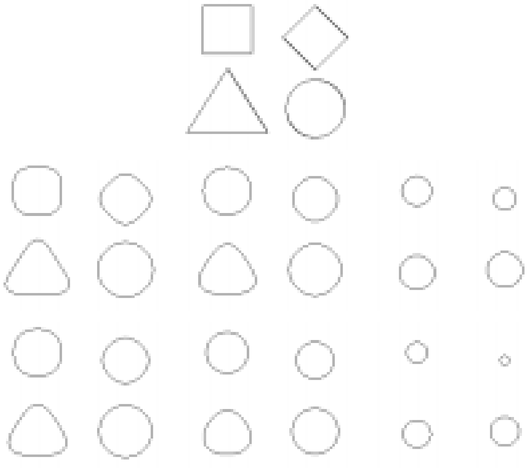


Figure 3.4: Middle row: Generator of symmetric random walk on $\boldsymbol{T}$ is shown to produce similar results with those in Bottom Row: Geometric Heat flow. The speeds of the two algorithms are different. The level set function is on a 191×221 grid, $\delta t = 0.25$.

feature/shape adapted flows.

3.4 A New Class of Flows

The insight gained from the tangential Brownian motion on a curve together with the normal angle $\theta(t, \boldsymbol{x})$, leads to the idea of constraining the Brownian motion at some specific orientation angles at each point $\boldsymbol{x}$. A natural modification of the geometric heat equation, based upon the stochastic framework presented in Section 3.3, is to construct an SDE weighted by a carefully chosen functional $h(\theta^n)$, ($h(.) \in C^\infty(\mathbb{R}^n)$) designed to capture specific features in an image, and we write locally

$$d\boldsymbol{X}(t) = \boldsymbol{T}^n(\boldsymbol{X}(t))\;(c^n(\boldsymbol{X}(t))h(\theta^n(\boldsymbol{X}(t)))\;dt + \sqrt{2}\,h(\theta^n(\boldsymbol{X}(t)))\;dB(t)).$$

Here, again, neglecting the drift motion and concentrating on pure diffusion, the Brownian motion in the tangent direction is being further constrained at some specific orientation values, i.e. at the zeros of the $h(\theta^n)$ function,

$$d\boldsymbol{X}(t) \approx \sqrt{2}\,\boldsymbol{T}^n(\boldsymbol{X}(t))\;h(\theta^n(\boldsymbol{X}(t)))\;d\tilde{B}(t)). \tag{3.12}$$

Constraining the diffusion of particles at points with specified orientations is aimed at extracting desired features of a contour as it is being smoothed. Such models are generated by the following class of PDE's, which directionally modify the geometric heat flow (3.2), and in this sense, generalize it by making the local generator of the diffusion SDE (3.12) conceivably arbitrarily selective,

$$\frac{\partial u(t,\boldsymbol{x})}{\partial t} = h^2(\theta(t,\boldsymbol{x}))\, u_{\xi\xi(t,\boldsymbol{x})}. \tag{3.13}$$

When applied to an image, this flow induces the following curve evolution on each iso-intensity contour $\boldsymbol{C}$

$$\frac{\partial \boldsymbol{C}}{\partial t} = h^2(\theta)\kappa \boldsymbol{N}\,. \tag{3.14}$$

3.4.1 Well-posedness of the generalized model

Proposition 2 *The corresponding PDE's (3.13) are well-posed.*

Proof 2 *The geometric heat equation which corresponds to the simplest case of this class with $h^2(\theta(t,\boldsymbol{x})) = 1,\ \forall t,\ \forall \boldsymbol{x}$, has been shown to be well-posed, and its existence and uniqueness properties may be found in [6, 31, 38]. The operator of the geometric heat equation is given by*

$$L[u] = \mathcal{L}[u] - \frac{\partial u}{\partial t} = 0 \tag{3.15}$$

where

$$\mathcal{L}[u] = \sum_{i,j}^{2} a_{ij} \frac{\partial^2 u}{\partial x_i\, \partial x_j} = \sin^2\theta\, u_{xx} - 2\sin\theta\ \cos\theta\, u_{xy} + \cos^2\theta\, u_{yy}, \tag{3.16}$$

is the principal part of the operator L. The matrix of coefficients $[a_{ij}]$ is positive semi-definite with the eigen values 1 and 0. If we multiply this matrix by a positive function, it remains positive semi-definite. Such elliptic-parabolic operators satisfy a maximum principle (see, for example, [119]). In our case we multiply by a non-negative function $h^2(\theta)$ which can be made strictly positive by adding a very small number, $\epsilon > 0$,

$$[h^2(\theta) + \epsilon]\mathcal{L}u > 0.$$

This results in a family of nonlinear parabolic equations each of which satisfies a strong maximum principle. Our operator is obtained in the limit as $\epsilon \to 0$.

3.4.2 Polygon yielding diffusions

The geometric heat equation is a rotationally invariant flow which evolves, as mentioned earlier, any shape into a circle [54]. It is the only rotationally invariant shape evolution in Euclidean space. If we wish to capture more general shapes (triangles, squares, etc...) it is only then natural to consider flows which are not rotationally invariant. Such a class is given by the form (3.14) when $h(\theta)$ is chosen to be other than a constant. If we are particularly interested in polygons, we may consider periodic functions (whose periodicity is dictated by the desired number of vertices) such as

$$h(\theta) = \begin{cases} \cos(n\theta) \\ \sin(n\theta), \end{cases} \tag{3.17}$$

leading to curve evolution equations of the form

$$\frac{\partial \boldsymbol{C}}{\partial t} = \cos^2(n\theta)\kappa\boldsymbol{N} \quad \text{or} \quad \frac{\partial \boldsymbol{C}}{\partial t} = \sin^2(n\theta)\kappa\boldsymbol{N}\,. \tag{3.18}$$

If we apply (3.18) to a convex shape, there will be $2n$ points on the curve which do not diffuse (corresponding to the zeros of $\cos(n\theta)$ or $\sin(n\theta)$) at equally separated rotations of the unit normal $\boldsymbol{N}$. As the unit normal moves further and further away from these angles, the diffusion increases. It hence makes sense to expect a curve to develop vertices (points of maximal curvature) at these points.

Lemma 1 *The angle of a unit normal does not change at points where the chosen function $h^2(\theta)$ vanishes. Those points, in turn, remain fixed for a short-time, and their speed remains at zero.*

Proof 3 *Assume that a family of curves $\boldsymbol{C}\,(t,p)$, where p is any parameter along the curve, evolves according to the evolution equation*

$$\frac{\partial \boldsymbol{C}}{\partial t} = \alpha(t,p)\boldsymbol{T} + \beta(t,p)\boldsymbol{N} \tag{3.19}$$

The evolution equation for the angle of the unit normal is given in [76] as

$$\frac{\partial \theta}{\partial t} = \frac{-1}{g}\left[\beta_p - \alpha\kappa g\right] \tag{3.20}$$

where $g = ||\boldsymbol{C}_p|| = \sqrt{\mathcal{X}_p^2 + \mathcal{Y}_p^2}$ is the length along the curve (metric). If we consider the case $\alpha = 0$ and $\beta = -h^2(\theta)\kappa$ (following the convention used by the authors in [76]), which corresponds *to the form of the deformation we proposed, the orientation evolution is governed by*

$$\begin{aligned}
\frac{\partial \theta}{\partial t} &= \frac{1}{g}\left[(h^2(\theta)\ \kappa)_p\right] \\
&= \frac{1}{g}\left\{2\ h(\theta)\ (h(\theta))_p\ \kappa + h^2(\theta)\ \kappa_p\right\} \\
&= \frac{1}{g}\ h(\theta)\ \left\{2\ (h(\theta))_p\ \kappa + h(\theta)\ \kappa_p\right\} \qquad (3.21)
\end{aligned}$$

Notice that $\frac{\partial \theta}{\partial t} = 0$ *for those points at which* $h(\theta) = 0$.

We note that in [76], the orientation of a curve is defined as the angle subtended by the tangent and the x-axis, whereas here we define θ as the angle subtended by the normal and the x-axis. There is, however, a complete equivalence in so far as the evolution equation of the angle θ is concerned.

In light of the above development, we can thus state that the zeros of the function $h(\theta)$ lead to fixed end points of curve segments. Fixing two end points, say a_1 and a_2 , we examine the evolution of curvature, whose general form is given by (in [76])

$$\frac{\partial \kappa}{\partial t} = -\frac{\partial^2 \beta}{\partial s^2} + \alpha \frac{\partial \kappa}{\partial s} - \beta \kappa^2,$$

where s is the arc-length parameter along the curve. When substituting $\alpha = 0$ and $\beta = -h^2(\theta)\kappa$ into this equation, we have

$$\begin{aligned}
\frac{\partial \kappa}{\partial t} &= [h^2(\theta)\kappa]_{ss} + h^2(\theta)\kappa^3 \\
\frac{\partial \kappa}{\partial t} &= [(h^2(\theta))_{ss}\kappa + 2(h^2(\theta))_s \kappa_s + h^2(\theta)\kappa_{ss}] + h^2(\theta)\kappa^3 \\
\frac{\partial \kappa}{\partial t} &= \underbrace{h^2(\theta)\ \kappa_{ss}}_{\text{diffusion term}} + \underbrace{h^2(\theta)\ \kappa^3 + (h^2(\theta))_{ss}\ \kappa + 2(h^2(\theta))_s\ \kappa_s}_{\text{reaction term}}
\end{aligned}$$

This clearly demonstrates that a regularizing diffusion takes place, since the multiplicative factor $h^2(\theta)$ never becomes negative (which would result in an ill-posed backward diffusion). In addition, we have the reaction term which is composed of functions of κ, κ^3, and κ_s.

We have hence shown that with fixed end points, a particular curve segment subject to the new evolution equation for the curvature shown above, results in a straight line as a final solution.

Now, we can state a theorem where we put our argument of convergence to regular polygons.

Theorem 1 *A convex curve* $\boldsymbol{C}$ *subject to the evolution* $\boldsymbol{C}_t = h^2(\theta)\kappa \boldsymbol{N}$ *will converge to an* M*-sided, regular polygon whose* M *vertices will be formed at those vanishing points of the function*

$h^2(\theta)$.

The proof of this theorem can be completed using the arc-length evolution equation

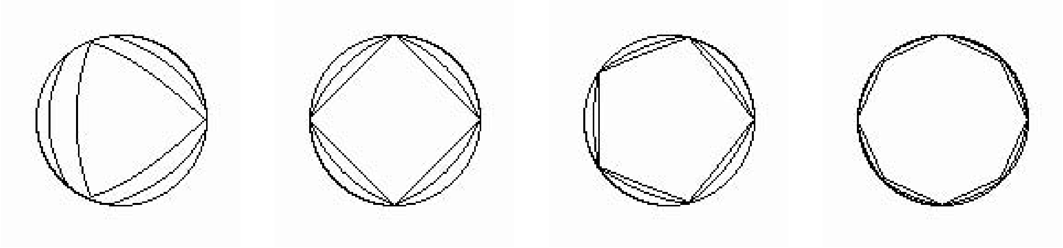
$$\frac{\partial L(\boldsymbol{C})}{\partial t} = -\int_{a_1}^{a_2} \kappa^2 h^2(\theta) ds,$$

where ds denotes the incremental arclength of $\boldsymbol{C}$. Since the integrand is strictly positive, we see that a curve will continue to shrink until curvature vanishes, that is the curve segment converges to a straight line between the end points a_1 and a_2. This, in conjunction with the above lemma completes the proof of the theorem.[1]

3.5 Experimental Results

3.5.1 Examples in Polygonization

To substantiate the stated theorem, and to intuitively illustrate our flows, we next present simulation results. In our experiments with contours, we use the narrowband implementation of the level set method developed in [2]. The time step is, $\delta t = 0.2$. Starting with a circular shape, the flow $\boldsymbol{C}_t = h^2(\theta)\kappa\boldsymbol{N}$ evolves it towards a specific polygon, i.e. it produces an $n-gone$ shape depending on the specific function $h(\theta)$. Several examples on morphing of a circle into different shapes are shown in Fig. 3.5. This is one potential application of the proposed flows in computer graphics, where the ability to morph a shape into a known other shape with an efficient algorithm is required for numerous applications. In addition to illustrating the propagation of the proposed flows in several snapshots, Fig. 3.5 also provides a quantitative and an objective means for characterizing



$h(\theta) = \sin(1.5\theta)$ $h(\theta) = \sin(2\theta)$ $h(\theta) = \sin(2.5\theta)$ $h(\theta) = \sin(4\theta)$

Figure 3.5: Morphing of a circle into different shapes by the given flows is demonstrated.

[1] Note added in Proof: We recently found out per Dr. Osher at UCLA that Peng, Osher, Merriman and Zhao, [116], have also independently proposed flows similar to those described in this thesis, albeit from a totally different perspective, and with a convective trend.

the performance of these algorithms in preserving corners. It can be observed that the position of the desired feature locations, i.e. the orientations at which the vertices of the final polygon are to be formed, are well preserved. There may inevitably be one-to-two pixel displacements due to numerical implementation effects, on account of the finite precision of the computations and the finite resolution of the grid (which affects almost all image processing algorithms).

The flows are also applied to a variety of convex shapes shown in part (a) of each figure: Fig. 3.6 and Fig. 3.7. In Fig. 3.6, the shapes in part (b) were obtained by using $h^2(\theta) = \cos^2(2\theta)$ via the following curve evolution

$$\boldsymbol{C}_t = \cos^2(2\theta)\kappa\boldsymbol{N}\ , \tag{3.22}$$

while the shapes in part (c) were obtained using $h^2(\theta) = \sin^2(2\theta)$. In both cases, we expect to obtain four-sided and regular polygons. The zeros of $\cos(2\theta)$ and the zeros of $\sin(2\theta)$ are however 45 degrees out of phase. As such, we see the evolved shapes in part (b) taking the form of a square, whereas the evolved shapes in part (c) take the form of a diamond, corresponding to a 45 degree rotation of the shapes in part (b). In Fig. 3.7, we see the effect of using different periods. The shapes in part (b) are obtained using $\boldsymbol{C}_t = \cos^2(3\theta)\kappa\boldsymbol{N}$, while the shapes in part (c) are obtained using $\boldsymbol{C}_t = \sin^2(1.5(\theta - \pi/2))\kappa\boldsymbol{N}$. In the first case, we expect 6 vertices, and in the second case we expect 3 vertices. Our expectations match the results shown in part (b) and (c), where we observe hexagonal and triangular shapes, respectively.

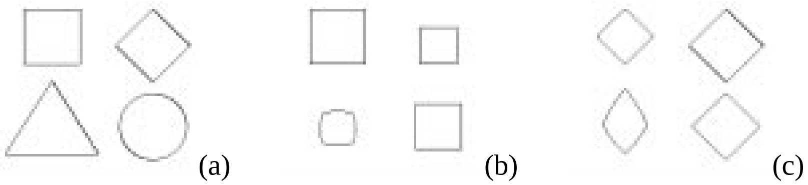


Figure 3.6: (a) Initial set of shapes (b) Flow $\boldsymbol{C}_t = \cos^2(2\theta)\kappa\boldsymbol{N}$ (c) Flow $\boldsymbol{C}_t = \sin^2(2\theta)\kappa\boldsymbol{N}$.

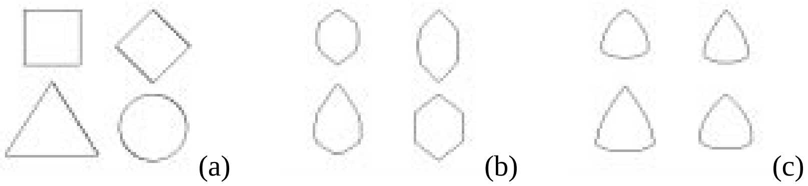


Figure 3.7: (a) Initial set of shapes (b)Flow $\boldsymbol{C}_t = \cos^2(3\theta)\kappa\boldsymbol{N}$, which tends to produce hexagons, (c)Flow $\boldsymbol{C}_t = \sin^2(1.5(\theta - \pi/2))\kappa\boldsymbol{N}$, which tends to produce triangle-like shapes.

These two figures also suggest an important potential application of the proposed flows, namely shape recognition. A typical scenario, would consist of an unknown shape, which is evolved with one of the polygonizing flows whose parameters are known a priori. The evolution which subsequently results in the least change in the input shape reveals the closest shape category the test shape may belong to.

Recall that we may also apply these flows to the level sets of an image in the same manner that the geometric heat equation may be applied. This gives rise to a family of anisotropic smoothing filters which, unlike the geometric heat equation, are not rotationally invariant. This feature can be useful in smoothing noisy images where corners and edges are priorly known to have certain orientations. These diffusions are effected by PDE's of the following form:

$$u_t = h^2(\theta)\nabla \cdot \left(\frac{\nabla u}{\|\nabla u\|} \right) \|\nabla u\|. \tag{3.23}$$

Note that the trigonometric expressions we have considered for $h^2(\theta)$ can be written in terms of the first derivatives of u, for example

$$\cos^2(2\theta) = \frac{(u_x^2 - u_y^2)^2}{(u_x^2 + u_y^2)^2} \quad \text{and} \quad \sin^2(2\theta) = \frac{(2u_x u_y)^2}{(u_x^2 + u_y^2)^2},$$

allowing one to implement the PDE without having to compute the orientation of the unit normal to each level curve. Note that these expressions involve only first order derivatives and therefore do not alter the quasi-linear structure of these second order flows.

The intended application of the proposed flows in this chapter, i.e. the smoothing of structures along the orientation of salient lines in both curves and images will be illustrated in the next two subsections, respectively.

3.5.2 Examples in Feature-Preservation

Feature-preserving properties as well as polygonal approximation properties of the proposed flows will be demonstrated in this section. We illustrate the idea of capturing different polygonal features of shapes by our proposed flows on the following examples.

The first example is a "chef" shape with both round and polygonal features as shown in Fig. 3.8. The geometric heat flow $\boldsymbol{C}_t = \kappa \boldsymbol{N}$, ($n = 0$), evolves these features into circles as shown in the second row of Fig. 3.8 for time points $t = 40, 80, 160$. Particularly, at $t = 160$, most parts of the shape turns into incomprehensible blob-like structures. In contrast to this, polygonal features of the chef like his nose, and tray, are preserved by the flow $\boldsymbol{C}_t = \sin^2(2\theta)\kappa \boldsymbol{N}$, ($n = 2$), which favors diamond-like structures (see third row of Fig. 3.8 for $t = 40, 80, 160$). Similarly, the flow $\boldsymbol{C}_t = \cos^2(4\theta)\kappa\boldsymbol{N}$, ($n = 4$), favors octagonal features as shown on the fourth row of Fig. 3.8, which is observed at chef's hat at all time points $t = 40, 80, 160$. The regularity of these flows is readily observed through the smoothness of the resulting shapes. When we view each row from left to right, we observe a progression from finer to coarser scale. The scale-spaces produced by our modified flows in the last two rows are visually more pleasing since corners are preserved, whereas in the row above we see them smoothed away by the pure geometric heat flow.

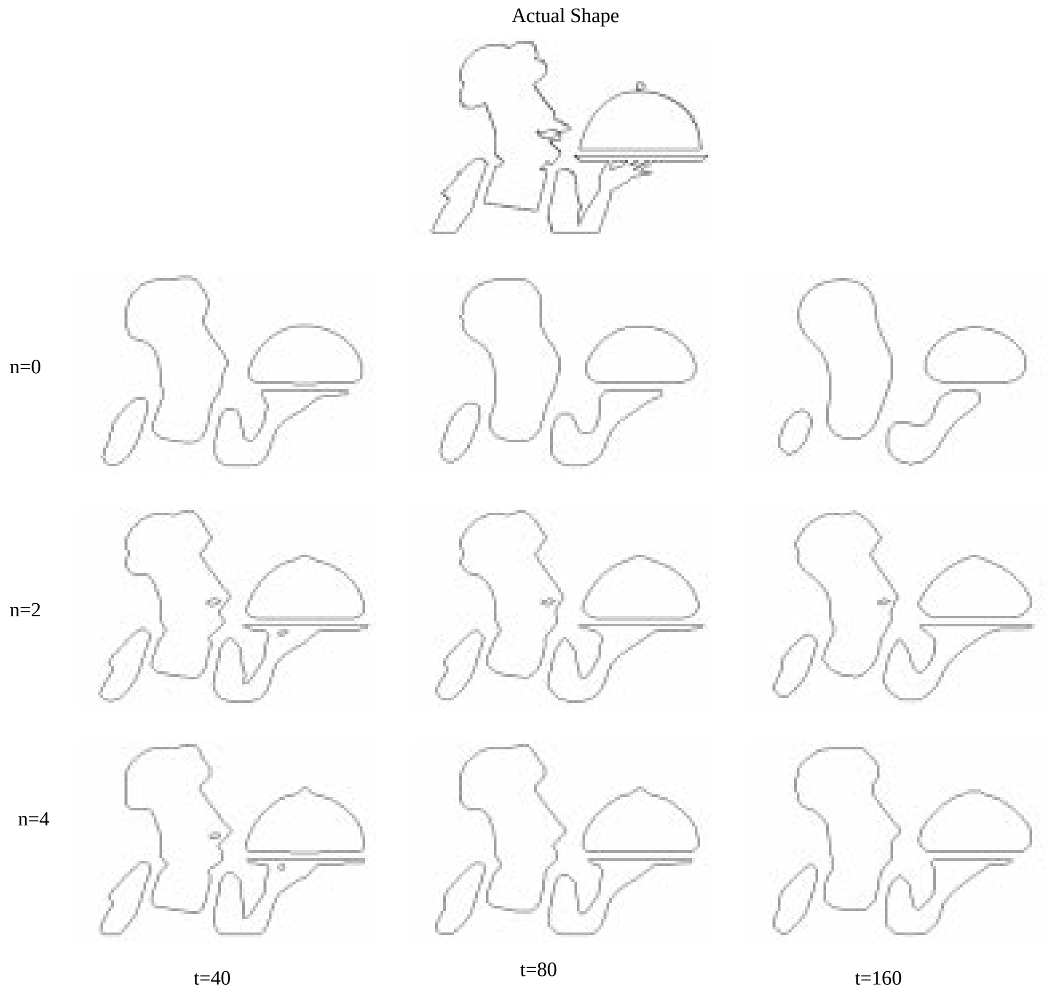


Figure 3.8: Each row corresponds to a curve evolution method with different n, 1^{st} row: $\boldsymbol{C}_t = \kappa\boldsymbol{N}$, 2^{nd} row: $\boldsymbol{C}_t = \sin^2(2\theta)\kappa\boldsymbol{N}$, 3^{rd} row: $\boldsymbol{C}_t = \cos^2(4\theta)\kappa\boldsymbol{N}$.

The second shape example is a fish which contains some fine detail structures as well as coarse features (Fig 3.9). The second row shows the result of the geometric heat flow $\boldsymbol{C}_t = \kappa\boldsymbol{N}$,($n = 0$), which smoothes away not only fine features but some coarse features as well (the fins for example). The results of the flow, $\boldsymbol{C}_t = \cos^2(2\theta)\kappa\boldsymbol{N}$,($n = 2$), are shown in the third row of Fig 3.9. In this case, rectangular features are preserved for longer periods throughout the evolution. Finally, the flow, $\boldsymbol{C}_t = \sin^2(4\theta)\kappa\boldsymbol{N}$, ($n = 4$), is depicted in the last row, preserving octagonal features as shown in the nose and the dorsal fins.

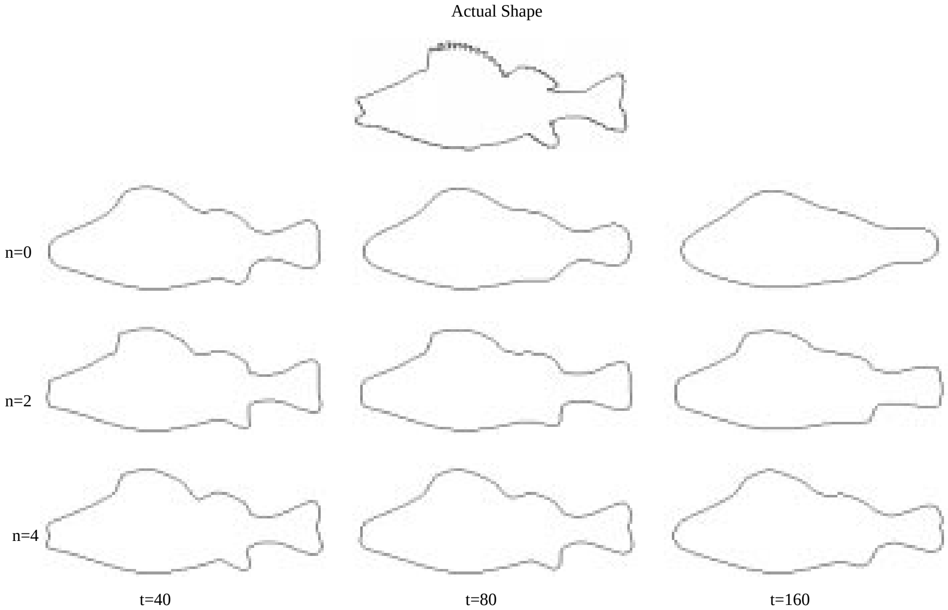


Figure 3.9: Each row corresponds to a curve evolution method with different n, 1^{st} row: $\boldsymbol{C}_t = \kappa\boldsymbol{N}$, 2^{nd} row: $\boldsymbol{C}_t = \cos^2(2\theta)\kappa\boldsymbol{N}$, 3^{rd} row: $\boldsymbol{C}_t = \sin^2(4\theta)\kappa\boldsymbol{N}$.

In the third example, we start with a noisy shape at time $t = 0$ shown in Fig. 3.10. This shape is evolved with the geometric heat flow $\boldsymbol{C}_t = \kappa\boldsymbol{N}$, ($n = 0$), the flow $\boldsymbol{C}_t = \sin^2(1.5(\theta - \pi/2))\kappa\boldsymbol{N}$, ($n = 1.5$), the flow $\boldsymbol{C}_t = \sin^2(2\theta)\kappa\boldsymbol{N}$, ($n = 2$), and the flow $\boldsymbol{C}_t = \cos^2(2.5(\theta - \pi/2))\kappa\boldsymbol{N}$, ($n = 2.5$), as shown in Fig. 3.10. The geometric heat flow at the top row quickly smoothes corners of the shape out, and at coarser scales, the shape loses all of its features. The initial shape converges to a circle in spite of the global feature of the plane being a polygonal shape. This motivates the application of the geometric heat flow with a $\sin^2(n\theta)$ factor, where n=1.5, and whose weak limiting shape is a triangle which intuitively matches the coarser form of the given plane shape. Similarly, for $n = 2$ and $n = 2.5$, different features of the shape are preserved, and persist over a much longer time period as can be observed from the column of shapes at $t = 400$. Note that the geometric heat flow result at the top quickly washes out any similarity to the actual shape, whereas the results of the other three flows preserve the global shape as well as some finer details on the wings, the tail, and the head part.

3.5.3 Examples with Grayscale Images

The proposed flows may also be applied to images in a straightforward fashion. For the case $h^2(\theta) = \cos^2(2\theta)$, all level sets of the image are driven to rectangles, thereby enhancing those features in an image. Such features can be found in contemporary buildings where one example in NCSU, Centennial Campus is shown in Fig.3.11(a). The part of the building image with an additive Gaussian noise is shown in Fig.3.11(b). The 2^{nd} row shows the results of the geometric heat flow $u_t = u_{\xi\xi}$ at $t = 10, 20, 50$. The noisy image at $t = 0$ is smoothed out very quickly at the expense of rounding off all the corners because the level sets of the image converge to circles. The 3^{rd} row shows the $u_t = \cos^2(2\theta)u_{\xi\xi}$ flow results at the same time points $t = 10, 20, 50$. Since the diffusion is constrained in order to drive image level sets to rectangles, the removal of noise is slower. However, the rectangular features still nicely appear after noise removal (see the image on the right), making it worthwhile to slow down the denoising effect of the geometric heat flow as deemed appropriate.

In Fig. 3.12, an experiment involving diamond-like shapes in the image taken from a poster on a wall is shown. In the middle row, rounding effects on diamond shapes performed by the geometric heat flow are clearly observed during the evolution of this image. The proposed flow, shown in the bottom row, takes the form $u_t = \sin^2(2\theta)u_{\xi\xi}$ for this particular shape, and particularly adapted to carrying out a shape-based smoothing which takes place at the boundaries of the diamonds. The slight blurring effect on the picture at continued application however is due to the interaction between consecutive level curves.

A photograph taken by pathfinder in mars, shown in Fig. 3.13, is argued to be a hexagon-shaped structure on mars' surface. The particular flow adapted to this shape is given by $u_t = \cos^2(3\theta)u_{\xi\xi}$, and the resulting images at the second column of the figure demonstrate a better smoothing performance at the boundaries of the hexagon when compared to the images in the first column processed by the geometric heat equation. From low scales to very large scales, the hexagon-adapted flow enhances and keeps on highlighting the related structure.

A noise contaminated Aerial image is shown in Fig. 3.14(a). The geometric heat equation (see 2^{nd} row, $t = 20, 40, 80$) sweeps away the shape information of the important details such as the city on the left bottom, the white bright rectangle on the right bottom, and the black feature at the top. The three images resulting from the $u_t = \cos^2(2\theta)\ u_{\xi\xi}$ flow, are quite sharp at the edges between both low and high contrast fields, therefore more useful in recognition of details as well as removing noise.

A last example is shown in Fig. 3.15, where windows and a roof of a section of a house are seen. On the left column, the results of the geometric heat flow $u_t = u_{\xi\xi}$ at times $t = 40, 80, 160$, are shown, and on the right those of $u_t = \cos^2(2\theta)u_{\xi\xi}$ at similar time steps. The noise is successfully removed by the geometric heat equation whose smearing effect on different regions into one another is also slow, at a cost of a problematic rounding off of corners. At time $t = 160$ for the result on the right bottom, approximately the same amount of noise as that of geometric heat equation at $t = 40$ is removed, and in addition to that the corners are still well-preserved.

3.6 Conclusions

In this chapter, also in [141], we have formulated a local stochastic view of a nonlinear filtering technique, namely the geometric heat equation. The theory of stochastic differential equations provides a microscopic view of a system, and through a local linearization of the nonlinear geometric heat equation we have provided an alternative macroscopic view of this equation. We then modified this macroscopic description to propose new flows that vanish at pre-defined directions. We showed that these flows, although rotationally non-invariant, are capable of smoothing along priorly known orientations of salient lines in both curves and images, leading to preservation of polygonal structures. In the context of curve evolutions, curves evolved with the new flows are morphed into a limiting polygonal shape, this approach may hence be relevant to shape-morphing applications in computer graphics. Another application of these polygonizing flows is in classification of a shape after its filtering via a certain set of these flows, and its identification according to the outputs of these different filters.

Our main idea of stopping the diffusion with a function of the normal angle θ, has an important characteristic that $h(\cdot)$ function should be periodic since the angle θ lives on the unit circle. We have presented in [140, 142], a variant of this idea where the modification of the geometric heat equation is accounted by a function that depends on the unit normal angle difference between neighboring points on the tangent line at a point of the curve. This also leads to a feature-driven smoothing of curves, (with different characteristics than the flows presented in this chapter), which has been used in filtering level sets of images, particularly of synthetic aperture radar imagery, and also in the target recognition of the latter.

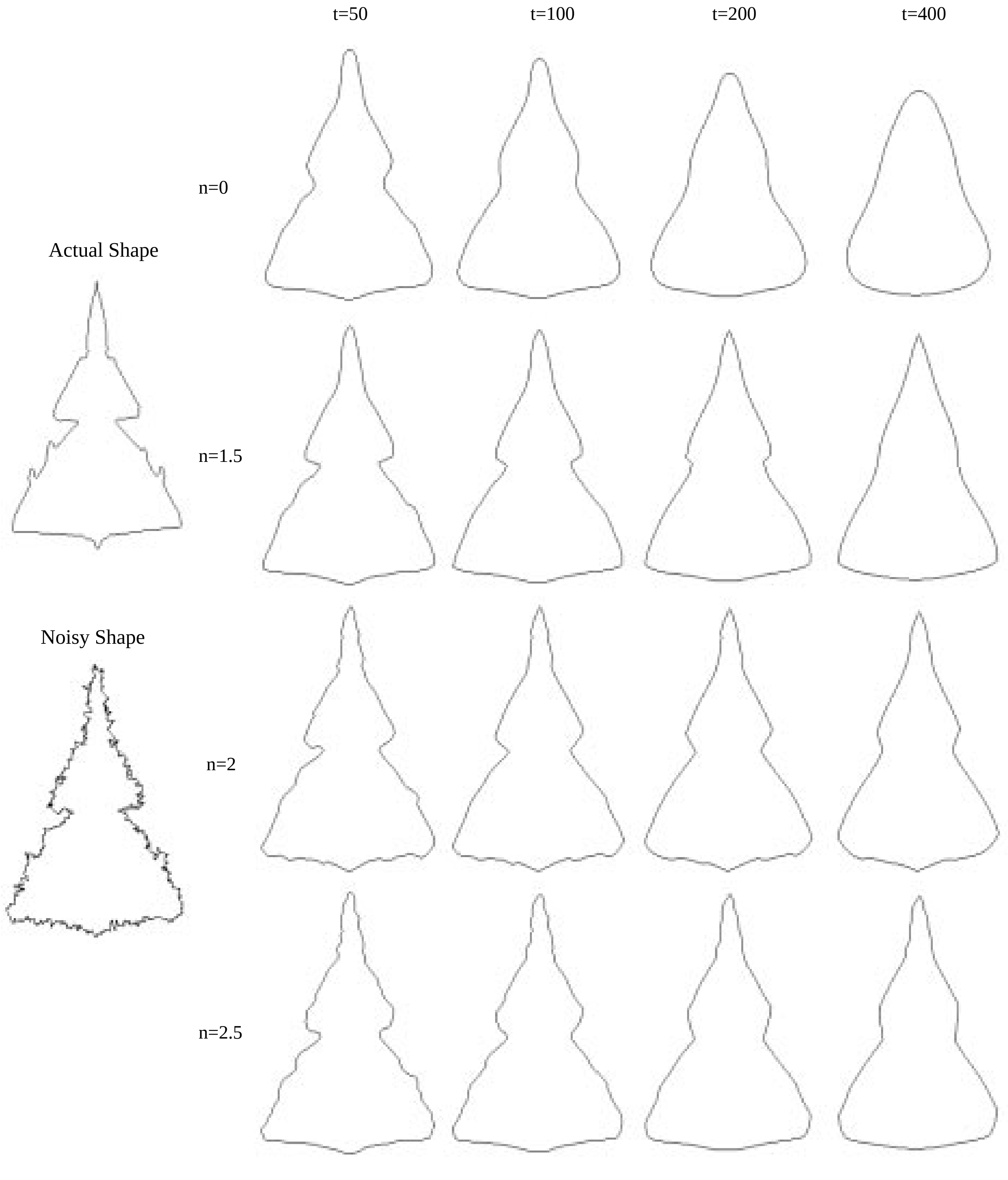


Figure 3.10: Each row corresponds to a curve evolution method with different n, 1^{st} row: $\boldsymbol{C}_t = \kappa\boldsymbol{N}$, 2^{nd} row: $\boldsymbol{C}_t = \sin^2(1.5(\theta - \pi/2))\kappa\boldsymbol{N}$, 3^{rd} row: $\boldsymbol{C}_t = \sin^2(2\theta)\kappa\boldsymbol{N}$, 4^{th} row:$\boldsymbol{C}_t = \cos^2(2.5(\theta - \pi/2))\kappa\boldsymbol{N}$.

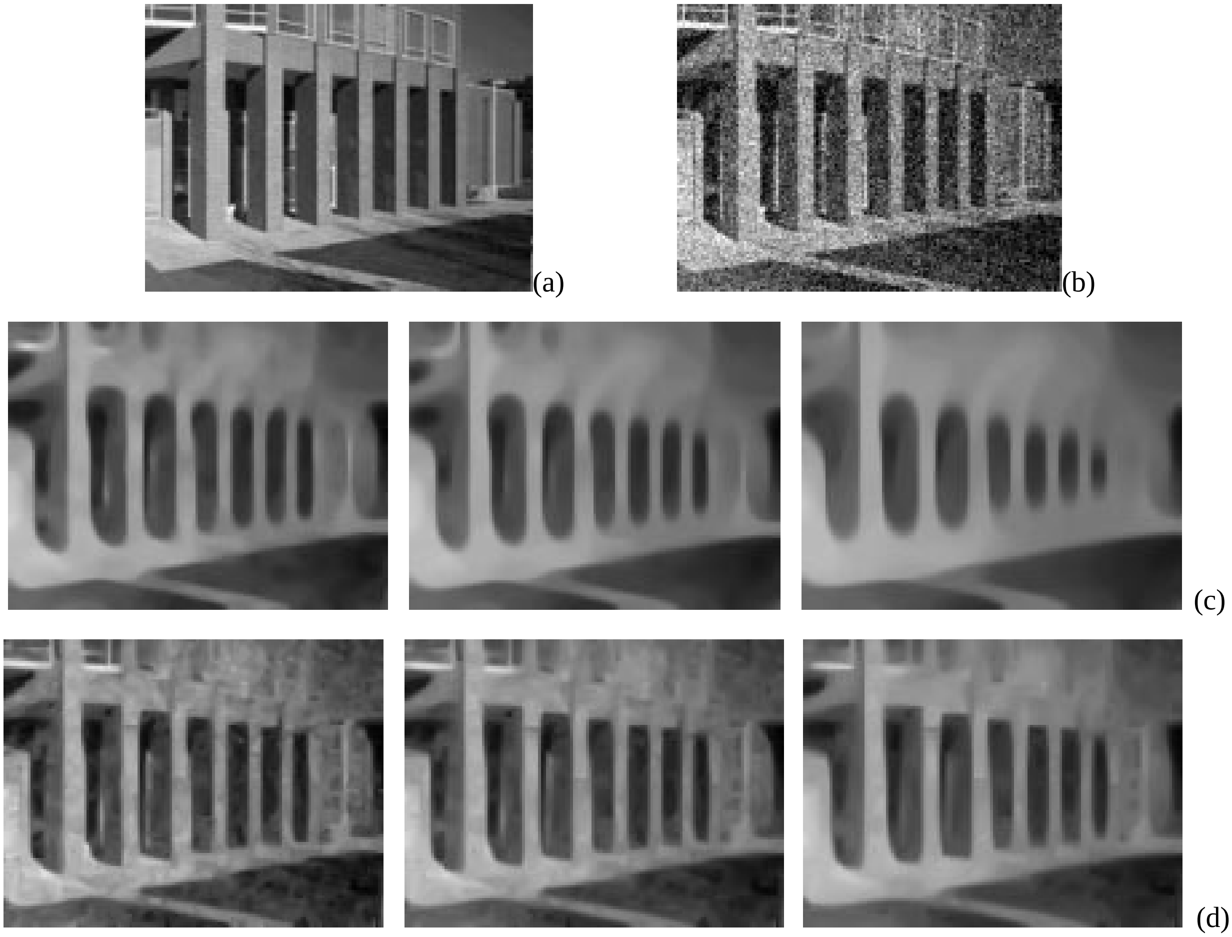


Figure 3.11: (a) Clean building image (b) noisy building image (c) Geometric heat flow $u_t = u_{\xi\xi}$ (left-right) $t = 10, 20, 40$ (d) Flow $u_t = \cos^2(2\theta)\, u_{\xi\xi}$ (left-right) $t = 10, 20, 40$.

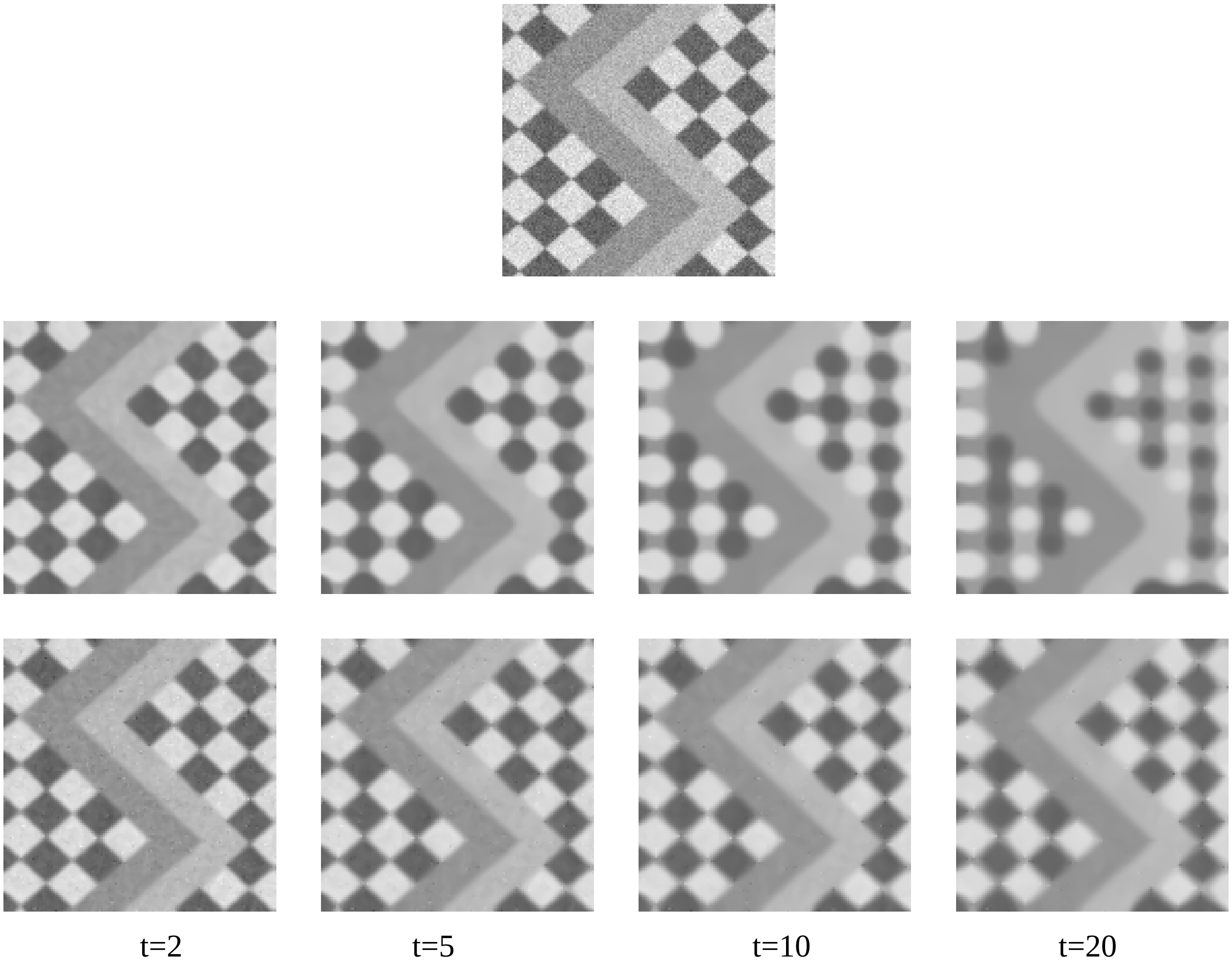


Figure 3.12: (Top) Diamonds image (Middle row) Geometric heat flow $u_t = u_{\xi\xi}$ (Bottom row) Flow $u_t = \sin^2(2\theta)\, u_{\xi\xi}$.

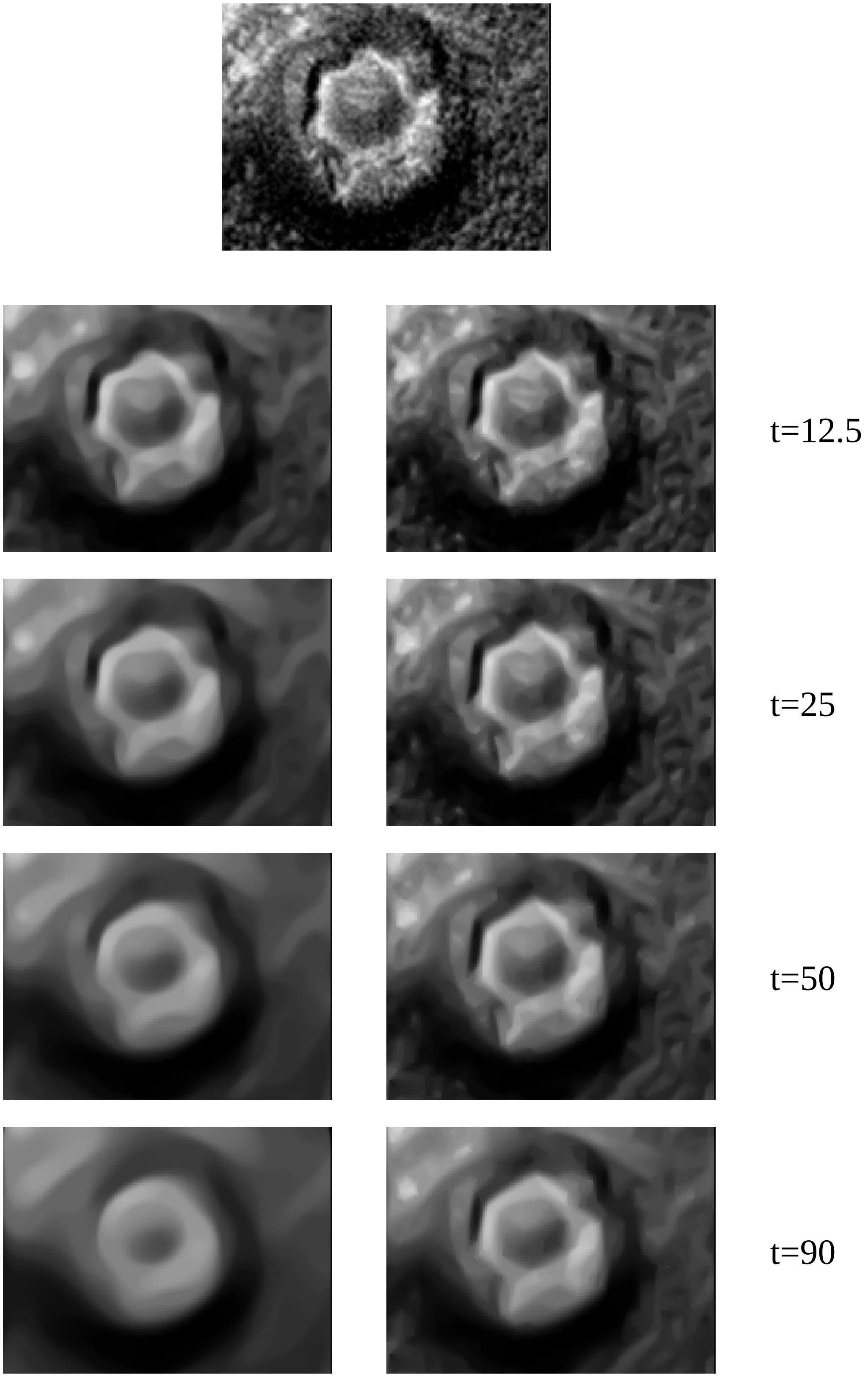


Figure 3.13: (Top) An image from mars pathfinder, (First column) Geometric heat flow $u_t = u_{\xi\xi}$, (Second column) Flow $u_t = \cos^2(3\theta)\ u_{\xi\xi}$. (Image: Origin NASA, exposed by and courtesy of N. Coombs & NPAAG 1998.)

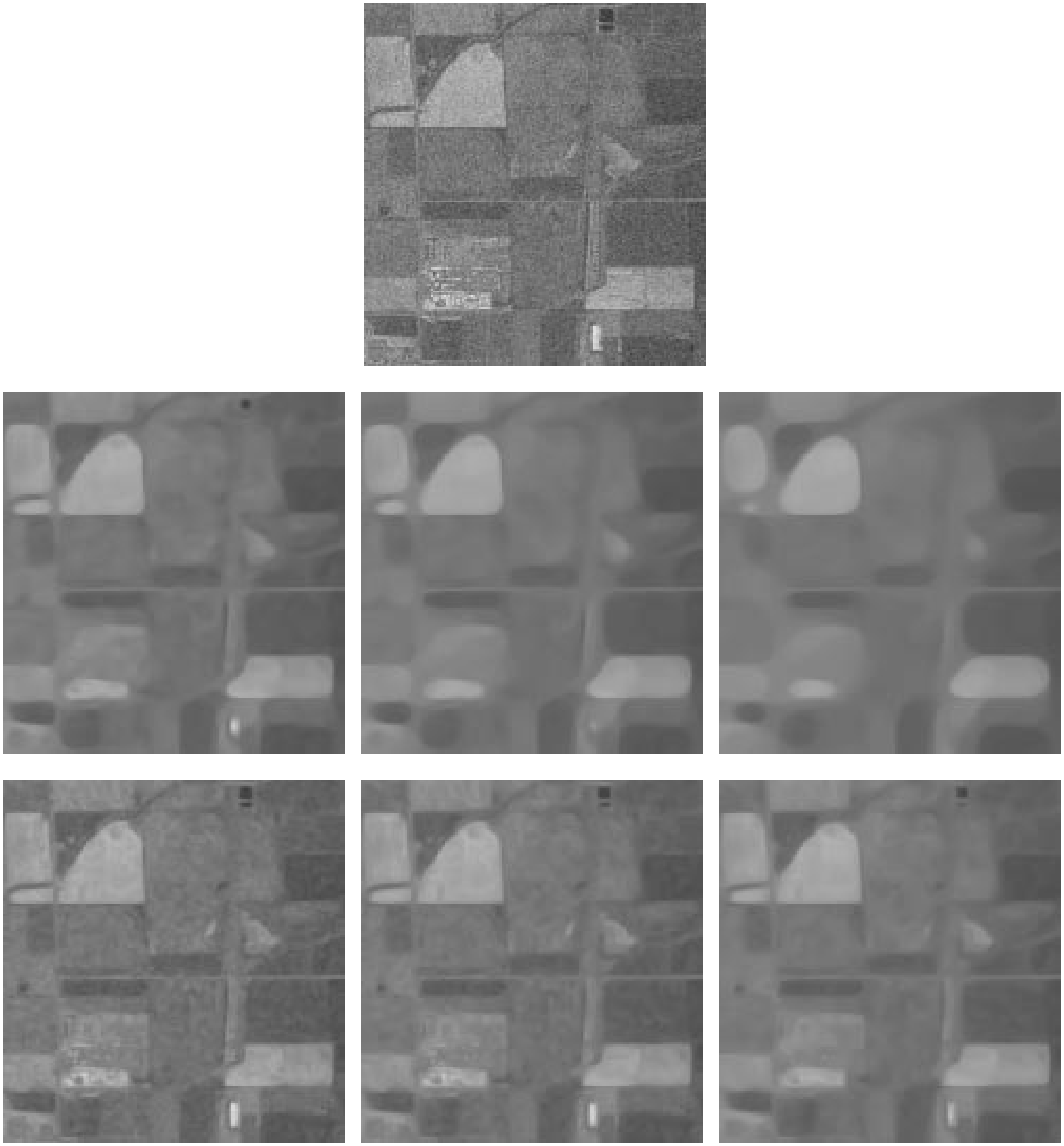


Figure 3.14: Top: Aerial image; 2^{nd} row: geometric heat flow $u_t = u_{\xi\xi}$, (left to right)$t = 20, 40, 80$; 3^{rd} row: flow $u_t = \cos^2(2\theta)u_{\xi\xi}$, (left to right) $t = 20, 40, 80$.

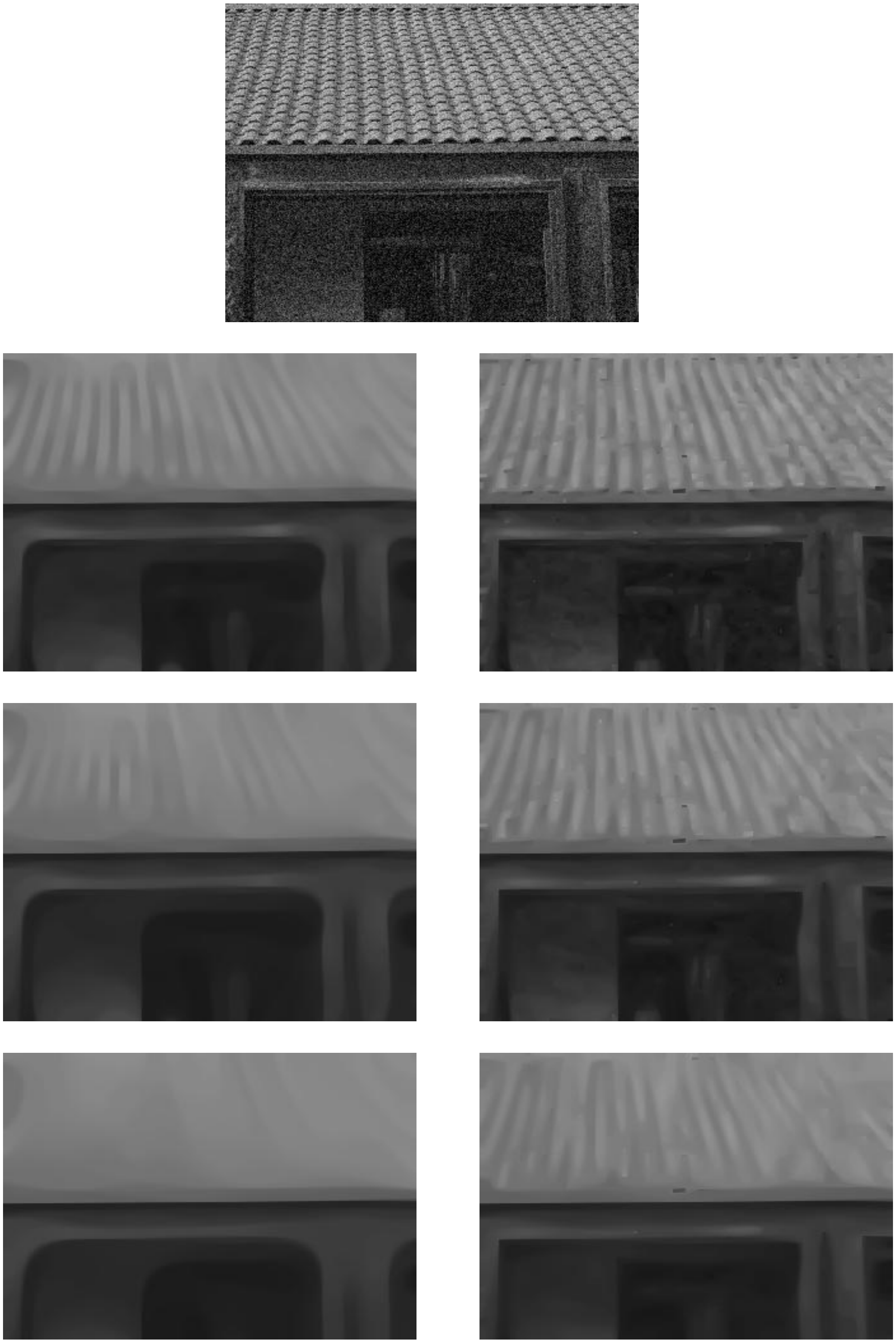


Figure 3.15: Top: House; Left: Geometric heat flow $u_t = u_{\xi\xi}$, (top to bottom) $t = 40, 80, 160$; Right: Flow $u_t = \cos^2(2\theta)\, u_{\xi\xi}$, (top to bottom) $t = 40, 80, 160$.

Chapter 4

Unsupervised Texture Segmentation by Information-Theoretic Active Polygons

This chapter introduces new ideas for image segmentation problem through new polygon-propagating equations in the active contour framework. Furthermore, adoption of an information-theoretic criterion as an energy functional of an active polygon is described for an unsupervised texture segmentation purpose. An overview on the general segmentation problem was given in Chapter 2.3. After briefly reviewing this literature in Section 4.1, an overview on texture analysis and modeling is presented in Section 4.2, along with our motivations, strategy, and the advantages of pursuing an active polygon model over adapting an active contour model. In Section 4.3, we derive the ODEs to obtain motion equations for polygon vertices. In Section 4.4, we introduce Jensen-Shannon criterion for evolution of a single contour and multiple contours with the goal of texture segmentation. In Section 4.5, a novel polygon regularizer to avoid degeneracy during a polygon propagation is introduced. To our knowledge, this type of global geometric regularizer has not been used even in related spline-based active contour models. Results, conclusions and discussions are given in Section 4.6.

4.1 Introduction

The problem of image segmentation refers to the partitioning of a given image domain into regions which are distinct in some sense. These, in turn, are expected to correspond to meaningful parts of objects in the image. Image segmentation is generally viewed as an essential first step in low level vision and as providing a mechanism for an automatic analysis of image contents. Some important applications include automatic target recognition, remote sensing, robot vision and guidance, automatic visual inspection in manufacturing processes, biomedical image analysis, tracking objects in motion. Segmentation is generally based on first determining features which delineate and reliably distinguish different regions. Some common features which discriminate the region characteristics are intensity, color, and texture. Selection of a “good” criterion is critical to capturing and evaluating the features which yield a partition of the image domain into distinct regions.

Earlier image segmentation methods were based on thresholding [112, 134] and on local filtering techniques [22, 96, 97], which make use of local information, and rely on high image gradient values to detect prominent boundaries between regions. This in turn makes them sensitive to noise and affects the continuity of the edge contours as a result. Region growing techniques [100] partition an image domain by merging regions according to some statistical criterion such as region variances, but their disadvantage is that they often generate irregular boundaries [153]. Snake [70] and active contour methods define energy functionals whose local minima comprise a set of solutions, e.g. boundaries of regions. The two main streams of thought are as follows:

- Inspired by the celebrated geometric heat equation [54], which may be obtained as a gradient descent flow of a Euclidean arc length, geometric active contour models [23, 25, 72, 93, 130], were developed as gradient flows of a modified Euclidean arc length. A modifying weight which is almost zero near edges, and almost one when it is far from the edges, can be chosen to be inversely proportional to the magnitude of the image gradient. These models are hence boundary-based, and are only sensitive to data near the curve (very local). This makes them prone to noise variability and to initial contour placement.

- To overcome this problem, region-based active contours, which use both local and global information were proposed [26, 27, 29, 121, 123, 149, 153]. These models mainly assume that the image consists of a finite number of regions, that are characterized by a pre-determined set of features or statistics such as means, and variances. An energy functional is constructed to pull these statistics apart. One advantage over the geometric active contours is that there is no need to calculate image gradients which are usually sensitive to noise, albeit at a cost of additionally imposed assumptions on the image.

The popularity of these active contour models was particularly due to an efficient implementation scheme via partial differential equations (PDEs), referred to as level set methods developed by Osher and Sethian for propagating interfaces [111, 127].

Global optimization approaches to image segmentation are based on energy functions minimizing different criteria such as Minimum Description Length (MDL) [87], and Bayesian [17]. Other methods were more hybrid in nature, such as the Region Competition method of Zhu and Yuille in [153], which adopt a generalized MDL criterion to essentially include the costs for coding the intensity of each pixel inside a region according to a prespecified probability distribution whose parameters are to be estimated. A region merging step is subsequently carried out while ensuring a decreasing total energy. A well-known mathematical model proposed by Mumford-Shah [103] combined a boundary extraction goal via an assumption of approximating an image by smooth functions in each region. Level-sets active contour implementation of this model have recently been introduced in [28, 137]. In fact region-based active contour models mentioned in the previous group can also be considered in this group, because their goal is also to extract boundaries of homogeneous regions within an image by minimizing an energy functional.

A particularly powerful cue in visual perception is the set of textural features. Texture segmentation, which is the task of parsing the image domain into a number of regions such that each region has the same textural properties is a challenging problem [68, 94]. The problem of unsupervised texture segmentation, which proceeds without a priori information about the textural characteristics of objects in a given image, is particularly challenging and remains largely an open research issue in computer vision. This along with its ubiquitous applications motivate further investigations into developing refined and sophisticated methods to accurately reflect and discriminate textured regions without supervision.

For textured images, defining a homogeneity measure is a mathematically challenging problem. We are going to propose a new texture segmentation approach which is an efficient improvement over the existing class of methods. Specifically we propose an information-theoretic measure, namely Jensen-Shannon divergence, which to our knowledge has not been used in the active contour framework, and seeks to distinguish and separate probability density functions of various regions in an image. This measure makes use of higher order statistics beyond mean and variance by a computationally efficient manner which does not require histograms.

4.2 Motivation and Previous Work

4.2.1 Related Work

For texture analysis and modeling, mainly two groups of approaches and their combinations have been developed:

- Visual system's decomposition of retinal images into a set of subbands, has been the source of inspiration for filtering techniques based on filterbanks of for instance Gaussians, and Gabor functions to characterize different attributes of a texture at different orientations and scales [68, 131]. Although multichannel filtering techniques have been shown to be efficient at capturing local spatial features, problems such as optimal choices of filters, and fusion of their outputs remain open.

- Statistical modeling approaches characterize texture images as arising from probability distributions on Markov Random Fields (MRF's) [13, 39, 51, 94], and hence mainly rely on a Markov property where each pixel is statistically dependent only on a certain set of neighboring pixels. The segmentation is achieved through a minimization of a maximum a posteriori (MAP) criterion of the observed image. An advantage of these approaches is that they yield the parameters of the underlying probability distribution which in turn affords one an ability to synthesize texture images by sampling. Limitations of commonly used MRF models, on the other hand, are due to the fact that only the first and second order statistics may tractably be used, while, it is widely recognized that many textures are strongly non-Gaussian regardless of the neighborhood size [154]. An analytical probability density for modeling clutter in natural images was proposed in [55].

- Methods which attempt to unify the afore-mentioned approaches have recently received attention. Zhu and Yuille [153] applied their model to texture segmentation where two local orientations of texture elements obtained through a Gaussian filtering of image gradient components are used as a multivariate texture image in their region competition model with underlying multivariate Gaussian distributions.

  Zhu, Wu, Mumford [154], also developed a probability model built on the texture features extracted by a set of filters aimed at capturing the properties of the texture at multiple scales and orientations. Their probability model is a maximum entropy distribution, which then includes the whole marginal distribution in this sense, and is hence more expressive than classical MRF's. This enables them to model non-Gaussian textures. We should however note that one of their main goals is texture synthesis which is a *different* goal than ours, which is a texture segmentation. This texture synthesis goal also bears a heavy computational burden, which we mean to bypass in our proposed approach.

Paragios and Deriche [115], in a supervised texture segmentation problem, also combined a filtering approach with a statistical view while accounting for boundary-based (geodesic) and region-based active contour frameworks. For a region-based attraction of the active contour, a posterior probability of each region with respect to the texture filter output image is used, exactly in the same way as in [153], with an MDL criterion, which, as well-known may also be stated as a MAP criterion. Their boundary module is a geodesic active contour model which utilizes an estimated boundary probability model to capture texture boundaries (analogous to capturing intensity boundaries with high image gradients). In contrast to our approach, this method requires a prior knowledge on the existing textures in the given image, and hence a pre-processing stage involving filter banks (e.g. Gabor), to estimate a Gaussian mixture probability distribution representing these pre-defined textures.

4.2.2 Synopsis of Proposed Approach

Our overall objective of object (homogeneous region) delineation in an observed image is highly motivated by classification and recognition applications. This in turn and in contrast to existing active contours, motivates a parsimonious and revealing representation such as that based on landmarks/vertices of an approximating polygon as we propose in this chapter. In addition to its parsimony, this representation is expected to accurately capture the prevailing texture and account for it in the course of the analysis.

Given an image, consisting of several differently textured and simply connected regions (or for our particular interest, textured objects), an approximation of their contours whose delineation is sensitive to their textural characteristics is achieved by adopting a polygonal model which we develop in Section 4.3, and by way of an information-theoretic measure, which we explore in Section 4.4 for propagating polygons. Combining these two goals yields our information-theoretic-based active polygon which is effective and appealing on account of its implicit inclusion of higher order statistics of the region data and integrated along the edges.

To motivate our investigation, we begin with a simple example of a synthetic image of two distinct intensity regions where the limitation of an active contour is readily seen as a result of excessive

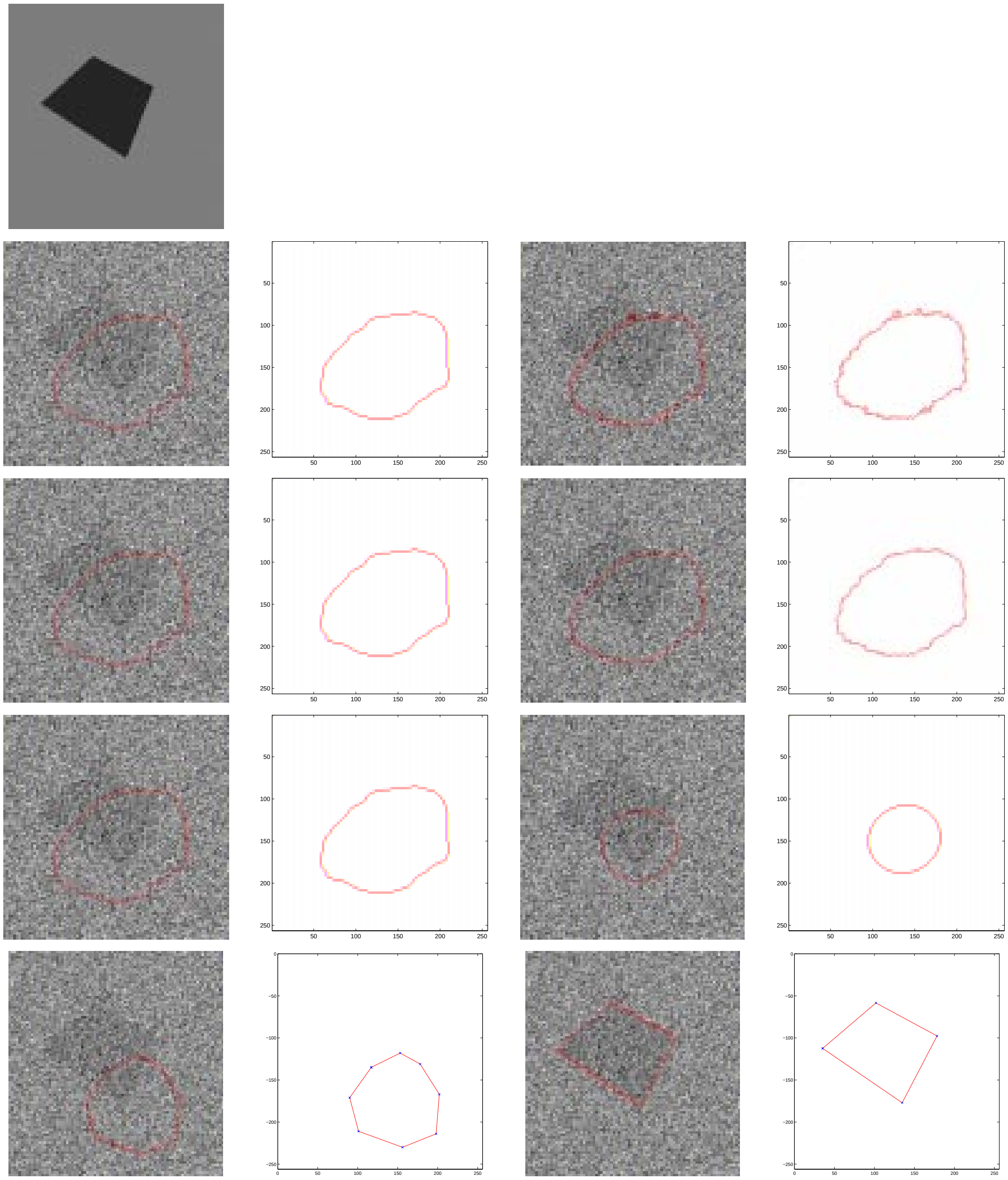


Figure 4.1: Motivational example where a simple binary image containing a quadrilateral (at the top) with additive uniform noise is to be segmented. Rows2-3-4 show a continuous active contour propagation with small, medium, large regularization respectively. The first two pictures in each column are the initial active contours on the image, and on a white background separately, and the last two pictures show what can possibly be obtained best in the continuous case. On the last row, an initial arbitrary active polygon (first two columns), is propagated successfully towards the target region, and the resulting polygon has only four vertices shown on the fourth column.

additive uniform noise, while a vertex-based polygonal evolution as will be described in detail easily captures the underlying object. This is illustrated in Fig.4.1. The essence of the active contour is to propagate an initial continuous curve towards the boundaries between two regions via a gradient descent flow of an energy functional with both image-based external energy, and a smoothness-constrained internal energy. Most active contour models, however, require significant regularization in the presence of significant noise in the image. With a little regularization on the continuous active contour model, (first row), the curve weaves around noise, with a medium regularization, (second row), still there is no way for an active contour to get out of local minima in this highly noisy case. With a very large regularization, the curve starts to converge to the shape, but the shrinking effect is too powerful, and the curve continues to shrink without sticking to the data, and eventually will collapse to a point. These three cases illustrate the continuum of effects from small to large regularization. We should note that this is based on a widely used arclength-based regularization, and that bad effects such as corners being rounded off, and the shrinking effects although not as prominent, will be seen in a similar way with rigidity terms (penalties on squared curvature). The contour would again ignore the data, and converge to a circle regardless of the shape of the object. By evolving an active polygon with a relatively small number of vertices, strong regularizing internal forces are no longer necessary to keep the contour from "breaking-up". A polygon laid on the same noisy image in Fig.4.1 (fourth row), propagates towards the boundaries with a greater resilience to noise, and results in a detection of the sharp corners. The target object is well-delineated, and the resulting polygon has only four vertices, accurately located at the corners of the object. While our introduced model is applied to a seemingly contrived simple example, nevertheless, illustrates the propagation of an active polygon through noise free of cost of sharp corners as well as its adaptivity and independence of initialization.

4.2.3 Strategy

To highlight our development strategy, we describe how the objectives stated at the outset may be achieved while clearly noting key conceptual differences with previously proposed techniques. The evolution of the vertices of a polygon according to an energy functional is clearly constrained by their set membership to the polygon and yields a set of coupled ordinary differential equations (ODEs) each corresponding to a particular vertex. We hasten to note here that our approach differs in an essential way from marker particle methods [127], or earlier snakes [70], in that the number of vertices of our active polygons is much smaller, and that the motion of vertices are obtained by integration of information over all edges rather than motion of individual particles. In marker particle methods, vertices behave independently under the influence of image forces, and coupling only comes from the internal forces. We note in addition that with our approach, these vertices naturally end up on important features such as corners on the boundary of an object. Rather than marker-particle based techniques, our approach is philosophically much more related to, but nevertheless quite different from, spline-based models, where the entire contour is represented by a few control points, such as [16, 45]. To prevent degeneracy during motion of the vertices (e.g. edges intersecting each other), we propose a global regularizer technique which uses electrostatics principles by viewing each edge of the polygon as a line of uniform charge. This particular point is also of interest in its own right as it introduces a global geometric regularizer rather than local geometric regularizers that have been popular.

To meet each of the objectives set out, we define an information-theoretic measure which, based on the so-called Jensen-Shannon divergence as an energy functional unfolds the textural information through the underlying probability distributions of the data, using easily computed image statistics. The formulation of such an energy functional in conjunction with a polygonal probing tool is followed by a gradient-based minimization procedure which evolves the vertices by way of a coupled set of ODEs just mentioned.

Note that Jensen-Shannon divergence [90], with Shannon entropy as a specific case [36], is a probabilistic difference measure, with properties of interest in various applications. For instance, unlike most other divergence measures, it can be generalized to a finite number of probability distributions, which makes it suitable to an image segmentation scenario with several target regions. For an image registration application, a similar divergence measure, based on a Renyi entropy [36], was proposed by [56, 58], referred to as a Jensen-Renyi divergence. Jensen-Shannon divergence was also applied to edge detection on regular or texture images in [53] by measuring the probabilistic differences between relative frequency of gray values in two halves of a small window that is slid over an image. Since it was developed for edge detection purposes, the result of this method yields unconnected edges, and contours, and requires further processing such as contour linking for a segmentation purpose. It hence differs from our approach which aims at finding a solid segmentation of objects in an image within a variational framework.

In contrast to Zhu and Yuille [153] for instance, our technique does not assume, for simplicity or for any other purpose, an underlying Gaussian model for texture regions. Our approach is hence statistically non-parametric, and image data drives the estimation of the divergence between the texture distributions as will be shown. A region-snake in [32], is developed for segmentation of images with only two regions, a target region and a background region, which are assumed as independent random variables, such an assertion, and particularly for optical images, required a prewhitening on the image. For segmentation of images of other modalities, such as ultrasound images and SAR images, authors in [33], and in contrast to our approach, assume particular prior densities which have explicit expressions, and then make use of a MAP criterion minimized by an iterative stochastic optimization via a polygon. The technique of Paragios and Deriche [115] was developed for a supervised texture segmentation, which requires a full knowledge of existing texture classes/regions, which may be multiply connected, thanks to the level-set implementation [111] of the PDEs. The approach we propose differs and may be viewed as a more challenging task, i.e. an unsupervised texture segmentation scenario, in which the only prior information is the number of different classes/regions which are assumed to be simply connected due to our simple polygon-moving ODEs.

Our approach in pursuing an active polygon model is further advocated next.

4.2.4 Advantages of Active Polygons over Active Contours

Active polygons enjoy some significant advantages and flexibility over existing techniques:

- A first advantage arises in texture segmentation. The system of ODEs integrate local image speed functions along two adjacent polygon edges when computing motion of each vertex. Thus, it turns local image measurements at points on the contour into global information, as these local measurements are integrated along the polygon edges. This makes our algorithm much more reliable and robust in capturing texture boundaries in contrast to continuous curves which are local in scope. Region-based methods also use global information inside and outside the curve, but their gradient flow incorporates local information, and pointwise on the curve, and is hence not amenable to speed functions for capturing higher-order statistics which can not be estimated from pointwise measurements, in any case.

- A reliable detection of sharp object corners is particularly important in recognition and classification of man-made objects, and the accurate performance of our algorithm is very accurate with no rounding effects at such important feature points, sets it apart from the continuous approaches.

- Emerging multimedia applications require targeted and specific data in large databases, which in turn entails a heavy demand for an efficient access and representation of visual objects in a scene. An efficient description of an object starts with its contour representation, which through a continuous curve requires a tremendous amount of information to be stored. If a contour is represented by a small number of vertices, this representation may later be compactly stored and provide a significant and informative amount of compression of the multimedia content. There is a vast amount of literature on obtaining polygonal approximations of continuous contours [42, 48, 66, 83]. The goal is often to reduce the amount of data (e.g. complex figures in cartography, geographical maps), and polygons are powerful approximators of shapes for use in later recognition stages [77], such as shape matching, and shape coding. Several techniques for shape coding such as [106, 151], entailed first extracting object boundaries and subsequently finding a polygonal approximation of the extracted contour. With goal of efficient compression, our technique, is hence advantageous in that it captures in one shot, a polygonal representation of objects in an image avoiding a typically lengthy search for meaningful locations, such as high curvature points, used in shape matching. The polygonal representation of an object from an image is only as good as the segmentation of the image. It is thus more efficient to invoke the polygonal representation directly in the course of the image segmentation process rather than afterwards. In short, our approach connects the final polygonal representation we wish to use directly to the image data.

- With a handful of vertices provided as a resulting object description, our method can efficiently be applied to visual image retrieval or indexing, and content-based description of multimedia data which are the core functionalities of MPEG-7 standard [30] .

- Another advantage of using ODEs rather than PDEs is a more efficient implementation. In the discretization, a significantly larger time step can be chosen to speed up the algorithm, whereas the PDEs usually require very small time steps for stable numerical implementations, particularly when strong regularizers, which are avoided by our model, are necessary. In addition, the identification of the interior and the exterior of a polygon is much easier than a continuous curve, thereby providing additional speed up in the calculations of statistics inside and outside of the curve.

- A particularly interesting property of the proposed approach is that a stochastic component may easily be incorporated, contrary to the continuous curve propagation by PDEs. One can indeed add a random perturbation component to the deterministic ODE component, such as a standard Brownian motion with a certain variance, in order to make the evolution more robust to noise effects which usually yield a convergence to a local extremum of the objective functional.

- Finally, we incorporate a novel global regularizer to the vertex motion equations using electrostatics principles. When the regularizer is the sole term acting on the polygon, it forces the polygonal edges and vertices to remain apart as imposed by the electric field induced at a vertex. We note that our proposed approach is to be distinguished from that of Bruckstein *et.al* [20], which, using a discrete form of curvature to account for only local geometry, evolves a polygon in the absence of an image term with a different intended application. Our global, rather than local, regularizer is more consistent in preserving local features such as corners.

4.3 Active Polygons

The dynamical equation of an active contour, typically follows the construction of an energy functional around a region which is subsequently minimized by a gradient descent flow. Our goal is to design flows to move a polygon by its relatively small number of vertices rather than a continuous active contour.

Let us consider a contour energy that consists of an integral inside and outside the active contour, where the integrand $f(x,y)$ consists of the function $f : \mathbb{R}^2 \to \mathbb{R}$. To explicitly invoke a contour $\boldsymbol{C} : [a,b] \subset \mathbb{R} \to \mathbb{R}^2$ around some region $R \subset \mathbb{R}^2$, we use the divergence theorem to write an integral over the interior of a curve as a contour integral

$$E(\boldsymbol{C}) = \iint_R f(x,y)\, dx\, dy = \oint_{\boldsymbol{C}=\partial R} \langle \boldsymbol{F}, \boldsymbol{N} \rangle\, ds,$$

where $\boldsymbol{N}$ denotes the outward unit normal to $\boldsymbol{C}$, ds the Euclidean arclength element, and where $\boldsymbol{F} = (F^1, F^2)$ is chosen so that $\nabla \cdot \boldsymbol{F} = f$. In what follows we will let $p \in [a,b]$ denote a fixed parameterization of the curve where $\boldsymbol{C}(a) = \boldsymbol{C}(b)$. We will indicate by v any variable whose variation affects the geometry of the curve. In the case of a polygon, v will denote a Cartesian coordinate of any vertex. In the case of a smooth curve, v will denote a curvilinear coordinate which varies along the normal direction at any point on the curve but remains constant along the curve itself. The key point, in either case, is that v and p constitute independent variables. We now rewrite the contour integral in terms of the parameter p,

$$E(\boldsymbol{C}) = \int_a^b \langle \boldsymbol{F}, \boldsymbol{N} \rangle \|\boldsymbol{C}_p\| \, dp = \int_a^b \langle \boldsymbol{F}, \boldsymbol{J}\,\boldsymbol{C}_p \rangle \, dp,$$

where $\boldsymbol{J} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, and $\boldsymbol{N}\,||\boldsymbol{C}_p|| = \boldsymbol{J}\,\boldsymbol{C}_p$. In order to determine the gradient flow associated with E, we compute the derivative with respect to v. After some manipulations, E_v can be obtained as [150, 153]

$$E_v(\boldsymbol{C}) = \int_a^b f \langle \boldsymbol{C}_v, \boldsymbol{J}\,\boldsymbol{C}_p \rangle \, dp. \tag{4.1}$$

In the case of a smooth curve $\boldsymbol{C}$, the variable v denotes, at each point on the curve, a coordinate which varies in the normal direction to the curve. Thus, if we are considering geometric evolutions $\frac{\partial \boldsymbol{C}}{\partial t}$ of the curve, we see from the final expression of E_v, that the gradient flow for $\boldsymbol{C}$ with respect to E is given by

$$\frac{\partial \boldsymbol{C}}{\partial t} = f \boldsymbol{N}. \tag{4.2}$$

Proceeding along a similar but slightly modified line of thought, we consider a closed polygon $\boldsymbol{V}$ as the contour $\boldsymbol{C}$, and with a fixed number of vertices $\{\boldsymbol{V}_1, .., \boldsymbol{V}_n\} = \{(x_i, y_i), i = 1, \ldots n\}$. We may parameterize $\boldsymbol{C}$ by $p \in [0, n]$ as

$$\boldsymbol{C}(p, \boldsymbol{V}) = L(p - \lfloor p \rfloor, \boldsymbol{V}_{\lfloor p \rfloor}, \boldsymbol{V}_{\lfloor p \rfloor + 1}) \tag{4.3}$$

where $\lfloor p \rfloor$ denotes the largest integer which is not greater than p, and where $L(t, \boldsymbol{A}, \boldsymbol{B}) = (1 - t)\boldsymbol{A} + t\boldsymbol{B}$ parameterizes between 0 to 1 the line from $\boldsymbol{A}$ to $\boldsymbol{B}$ with constant speed, ($\boldsymbol{A}$ and $\boldsymbol{B}$ denote the end points of a polygon edge). Note that the indices of $\boldsymbol{V}$ should be interpreted as modulo n so that $\boldsymbol{V}_0$ and $\boldsymbol{V}_n$ denote the same vertex (recall $\boldsymbol{C}$ is a closed curve). Finally, note that $\boldsymbol{C}_p$ is defined almost everywhere (where $p \neq \lfloor p \rfloor$) by

$$\boldsymbol{C}_p(p, \boldsymbol{V}) = \boldsymbol{V}_{\lfloor p \rfloor + 1} - \boldsymbol{V}_{\lfloor p \rfloor}. \tag{4.4}$$

Proposition 3 *Using the above parameterization $\boldsymbol{C}_p(p, \boldsymbol{V})$ in Eq. (4.1), we obtain the first variation of the energy functional E as*

$$E_v = \int_0^n f\big(L(p - \lfloor p \rfloor, \boldsymbol{V}_{\lfloor p \rfloor}, \boldsymbol{V}_{\lfloor p \rfloor + 1})\big) \left\langle \boldsymbol{C}_v, J(\boldsymbol{V}_{\lfloor p \rfloor + 1} - \boldsymbol{V}_{\lfloor p \rfloor}) \right\rangle dp, \tag{4.5}$$

and the minimization of E is achieved by a gradient descent flow given by a set of ODEs for each *vertex* $V_k, \;\; k = 1, ..., n$*, as (see Appendix B)*

$$\frac{\partial \boldsymbol{V}_k}{\partial t} = \int_0^1 p f\big(L(p, \boldsymbol{V}_{k-1}, \boldsymbol{V}_k)\big)\, dp\, \boldsymbol{N}_{k,k-1} + \int_0^1 (1-p) f\big(L(p, \boldsymbol{V}_k, \boldsymbol{V}_{k+1})\big)\, dp\, \boldsymbol{N}_{k+1,k}, \quad (4.6)$$

where $\boldsymbol{N}_{k,k-1}$ *(resp.* $\boldsymbol{N}_{k+1,k}$*) denotes the outward unit normal of edge* $(\boldsymbol{V}_{k-1} - \boldsymbol{V}_k)$ *(resp.* $(\boldsymbol{V}_k - \boldsymbol{V}_{k+1})$*).*

Written for each vertex, these equations are a set of coupled ordinary differential equations to be simultaneously solved. Intuitively, any of these equations may concisely be re-written as

$$\frac{\partial \boldsymbol{V}_k}{\partial t} = \tilde{f}_{k,k-1} \boldsymbol{N}_{k,k-1} + \tilde{f}_{k+1,k} \boldsymbol{N}_{k+1,k} \quad (4.7)$$

integrates the information, which is obtained from image values along two adjacent edges $(\boldsymbol{V}_{k-1} - \boldsymbol{V}_k)$, and $(\boldsymbol{V}_k - \boldsymbol{V}_{k+1})$, combined with the global image statistics (depending on the function f), into two overall speed functions $\tilde{f}_{k,k-1}$, and $\tilde{f}_{k+1,k}$ in order to move the vertex $\boldsymbol{V}_k$. This system of ODEs hence effectively implements a coupled motion of all the vertices of a polygon. In addition, this integration procedure, provides improved robustness to noise. Another advantage of the flow in Eq.(4.6) is the reduction of the dimension of contour propagation problem from a theoretically infinite one to roughly on the order of $3-30$ vertex points, depending on the complexity of an object boundary. This also clearly highlights the differences mentioned earlier between active polygons and marker particle methods. One may note the significantly reduced number of well separated vertices which have to be propagated. In addition, the motion of each vertex being based on a weighted combination of the unit normals only at the polygon's edge points, implies that there is no need for a unit normal to be defined at a vertex. As a result, the polygonal contour formed by these vertex locations $(V_1, ..., V_n)$ need not be differentiable. A generic image-based term, indeed a speed function f, can be used in the derived ODEs describing the motion of our active polygons, hence making them quite flexible.

For illustration, we apply our active polygon model in Eq.(4.6) to an image using simple mean statistics for the foreground and the background. For a given image, with a simple triangle shape as the target, the propagation of an active polygon with different initializations toward the target shape are shown in each row of Fig. 4.2. The edges of the active polygon align themselves along the edges of the target object, with the influence of the weights determined by the data term, i.e. the two weighted integrals along two neighboring sides at each vertex. The result of our algorithm for

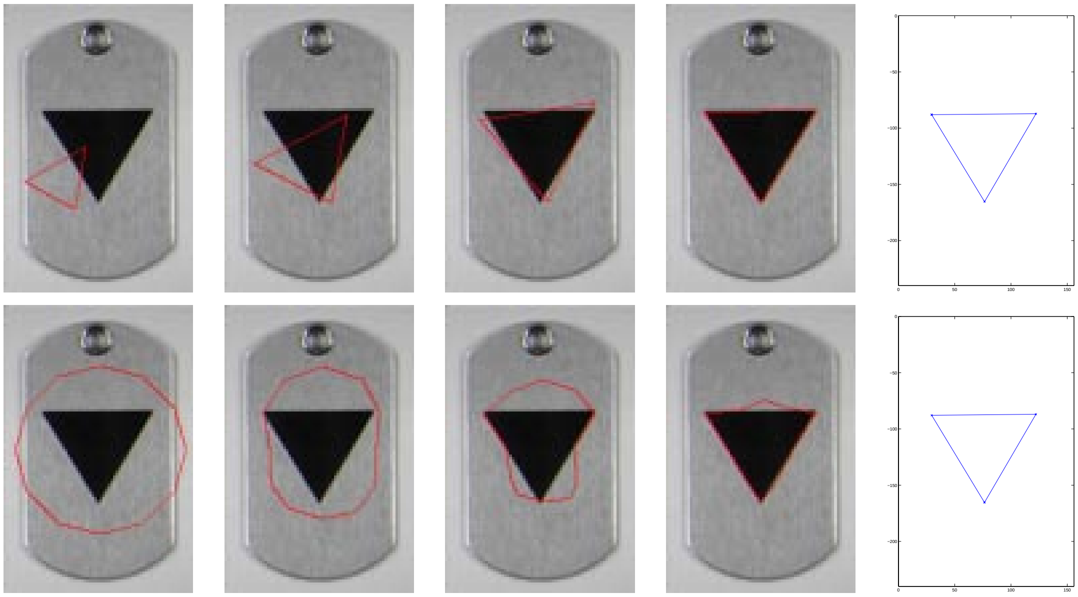


Figure 4.2: We demonstrate flow (4.6) for two different initial active polygons.

two different initializations, given at the right of each row, is indeed the location of 3 vertices of the target triangle. Implementation issues along with the initialization are discussed in Section 4.6.1.

4.4 An Information-Theoretic Criterion

It is commonly assumed in region-based active contours that an image is piecewise constant. Several techniques based on utilizing the first and second order statistics have been proposed. While these techniques are quite adequate for Gaussian data, they are highly insufficient to capture the underlying information in non-Gaussian data which include almost all textures. Towards that end, we use an information theoretic measure which not only captures the higher order information of non-Gaussian data, but provides a probabilistic disparity measure among N data populations. Hence, we consider a decision problem with N classes $c_1, \ldots, c_N$ with prior probabilities, $\boldsymbol{a} = (a_1, \ldots, a_N)$ such that $\sum_{i=1}^{N} a_i = 1$. Based on the Shannon entropy H, the so-called generalized Jensen-Shannon divergence [90] among N probability densities is

$$JS_a = H\left(\sum_{i=1}^{N} a_i p_i(\xi)\right) - \sum_{i=1}^{N} a_i H(p_i(\xi)) \tag{4.8}$$

where $p_i(\xi)$ denotes the probability density of the i^{th} class in a region. One of the major features of the Jensen-Shannon divergence is that different weights a_i may be assigned to the relevant distributions according to their importance.

Our ability to better capture the underlying statistics of regions using the entropy-based symmetric divergence measure, suggests that this may play a key role in constructing an energy functional whose optimization would yield a polygonal flow to well delineate differently textured regions. To proceed, and for the sake of efficiency, we resort to a fast numerical estimation of the densities which in turn facilitates the estimation of the entropies and hence of the divergence measure. We chose to adopt a first order approximation of a density which achieves the maximum entropy solution, and which, in turn, is used in approximating the entropy expression as proposed by Hyvarinen in [64].

On a region delineated by an active contour, we assume a scalar random variable (r.v.) $\boldsymbol{\xi}$ on a given set of intensity values Ξ, $\xi : \Xi \subset \mathbb{R} \to \mathbb{R}^+$, and the available information on the density of the r.v. ξ is given by

$$\int_{\Xi} p(\xi) G_j(\xi) d\xi = u_j, \quad \text{for } j = 1, .., m, \tag{4.9}$$

where the estimates u_j are the expectations $E\{G_j(\boldsymbol{\xi})\}$ of m known independent functions $\{G_j(\cdot)\}$ of $\boldsymbol{\xi}$. Note that there is no model assumption for the random variable $\boldsymbol{\xi}$, however, the distribution which has the maximum entropy and which is also compatible with the measurements in Eq. (4.9) is sought [36, 69]. This problem has been widely studied, and the form of the maximum entropy distribution has been derived in [36, 69]. The density $p_o(\xi)$ which satisfies the constraints (4.9) and has maximum entropy among all such densities is of the form $p_o(\xi) = Be^{\sum_{j=1}^{m} b_j G_j(\xi)}$ where B and b_j are constants that are determined from the constraints. A first order approximation for this maximum entropy density denoted by $\hat{p}(\xi)$, is given by [64]

$$\hat{p}(\xi) = \phi(\xi) \left(1 + \sum_{j=1}^{m} u_j G_j(\xi) \right) \tag{4.10}$$

where $u_j = E\{G_j(\boldsymbol{\xi})\}$, and $\phi(\xi)$ is the standard Gaussian density $\phi(\xi) = \exp(-\xi^2/2)/\sqrt{2\pi}$. ($E\{\cdot\}$ is the expectation of a random variable). In practice, the u_j's are estimated as the corresponding sample averages of the $G_j(\xi)$, i.e. we compute measurements in a region R of an image function $I : \mathbb{R}^2 \to \mathbb{R}$ by $u_j = \frac{1}{|R|} \sum_{(x,y) \in R} G_j(I(x, y))$, where $|R|$ is the number of pixels in R. A simple approximation of the entropy functional is subsequently found upon substituting the approximate density $\hat{p}(\xi)$ in

$$H(\hat{p}(\xi)) = - \int_{\Xi} \hat{p}(\xi) \log \hat{p}(\xi) d\xi \quad \approx H(\nu) - \frac{1}{2} \sum_{j=1}^{m} u_j^2, \tag{4.11}$$

where $H(\nu) = \frac{1}{2}(1 + \log(2\pi))$ is the entropy of a standardized Gaussian variable (see [64] for details). This result implies that a first order approximate maximum entropy of a given distribution is found by calculating how far away it is from that of the standard Gaussian density.

4.4.1 Evolution of A Single Active Contour

With a single contour, the image domain $\Omega \subset \mathbb{R}^2$ is split into two regions, namely a region inside the contour, call it R_u, and a region outside the contour, $\Omega \backslash R_u$. Exploiting the approximation in Eq. (4.11), we define an energy functional whose optimization yields the evolution of our active contour based on a divergence measure defined in Eq. (4.8). The new energy functional for two regions, is

$$\begin{aligned} JS_{a,2} &= H(a_1\, \hat{p}_1(\xi) + a_2\, \hat{p}_2(\xi)) - a_1\, H(\hat{p}_1(\xi)) - a_2\, H(\hat{p}_2(\xi)) \\ &= -\int_{\Xi} (a_1\, \hat{p}_1(\xi) + a_2\, \hat{p}_2(\xi)) \log(a_1\, \hat{p}_1(\xi) + a_2\, \hat{p}_2(\xi)) d\xi \\ &\quad + a_1 \int_{\Xi} \hat{p}_1(\xi) \log(\hat{p}_1(\xi)) d\xi + a_2 \int_{\Xi} \hat{p}_2(\xi) \log(\hat{p}_2(\xi)) d\xi, \end{aligned} \tag{4.12}$$

Here the expression $H(a_1\, \hat{p}_1(\xi) + a_2\, \hat{p}_2(\xi))$ may similarly to $H(p_1(\xi))$ be approximated by substituting the approximate density expression in Eq.(4.10) to yield

$$\begin{aligned} H(a_1\, \hat{p}_1(\xi) + a_2\, \hat{p}_2(\xi)) &\approx -\int_{\Xi} \left(a_1\, \phi(\xi)[1 + \sum_{j=1}^{m} u_j G_j(\xi)] + a_2\, \phi(\xi)[1 + \sum_{j=1}^{m} v_j G_j(\xi)] \right) \\ &\qquad \cdot \log \left(a_1\, \phi(\xi)[1 + \sum_{j=1}^{m} u_j G_j(\xi)] + a_2\, \phi(\xi)[1 + \sum_{j=1}^{m} v_j G_j(\xi)] \right) d\xi \\ &= -\int_{\Xi} \phi(\xi) \left(1 + \sum_j (a_1\, u_j + a_2\, v_j)\, G_j(\xi) \right) \\ &\qquad \cdot \log \phi(\xi) \left(1 + \sum_j (a_1\, u_j + a_2\, v_j)\, G_j(\xi) \right) d\xi \\ &\approx H(\nu) - \frac{1}{2} \sum_{j=1}^{m} (a_1\, u_j + a_2\, v_j)^2. \end{aligned} \tag{4.13}$$

This shows that the same first order entropy approximation holds for the sum of densities. Denoting measurements inside the contour as u_j, and those outside as v_j, for $j = 1, .., m$, and using the approximate entropy expressions (4.11), (4.13) in the energy functional Eq.(4.12), it may be approximated as

$$\begin{aligned}
\widehat{JS}_{a,2} &= H(\nu) - \frac{1}{2}\sum_{j=1}^{m}(a_1\, u_j + a_2\, v_j)^2 - a_1 \left(H(\nu) - \frac{1}{2}\sum_j u_j^2 \right) - a_2 \left(H(\nu) - \frac{1}{2}\sum_j v_j^2 \right) \\
&= \frac{1}{2}\sum_{j=1}^{m}\left(-(a_1\, u_j + a_2\, v_j)^2 + a_1\, u_j^2 + a_2\, v_j^2 \right) \\
&= \frac{1}{2}\, a_1\, a_2 \sum_{j=1}^{m}(u_j - v_j)^2 \qquad (4.14)
\end{aligned}$$

by noting that $a_1 + a_2 = 1$.

Note that on account of the higher order nature of the coefficients u_j and v_j (i.e. higher order than first and second moments), the proposed energy functional subsumes the previously proposed techniques based on the first and second order statistics. Those may in fact be shown to be particular cases of the above. This may also be justified by the fact that non-Gaussian densities may be expanded and that their higher order cumulants reflect the degree (or the lack thereof) of skewness or kurtosis of a density relative to a normal [71], hence $G(\xi) = \xi^3$, and $G(\xi) = \xi^4$, would be particular choices for the measurement functions G_j. Other choices of these functions; $G_1(\xi) = \xi e^{-\xi^2/2}$ as an odd function to measure asymmetry (analogous to the third moment as a measure of skewness), and $G_2(\xi) = |\xi|$, or $e^{-\xi^2/2}$ as choices of even functions (analogous to the fourth moment as a measure of sparsity, bimodality, relative to a Gaussian density), are given in [64, 65]. The idea behind using expectations of odd and even functions of the data is thus similar to attempts to characterize a density by the first few moments (usually order is less than 4). The choice of priors a_i for each density is explained next:

- If we assign constant priors a_1 and a_2 (e.g. equal priors $a_1 = a_2 = 0.5$) for both of the regions, the constant $a_1 a_2/2$ multiplying the energy functional in Eq. (4.14) has no effect in terms of its first variation, and the gradient descent flow of the active contour that minimizes the energy functional $E = -\widehat{JS}_{a,2} = -\frac{1}{2}a_1\, a_2 \sum_{j=1}^{m}(u_j - v_j)^2$ may be obtained by taking its first variation as

  $$\frac{\partial \boldsymbol{C}}{\partial t} = \nabla \widehat{JS}_{a,2} = \sum_{j=1}^{m}(u_j - v_j)(\nabla u_j - \nabla v_j), \qquad (4.15)$$

  where a continuous active contour $\boldsymbol{C}$ is utilized. As we have noted in the previous section, the resulting flow can be directly applied to an active polygon instead of a continuous active contour. A measurement or a constraint on a region distribution; u_j (resp. v_j) for region R_u (resp. R_v), is given by

$$u_j = \frac{\int_{R_u} G_j(I(x,y))dxdy}{|R_u|} \;,\; v_j = \frac{\int_{\Omega\backslash R_u} G_j(I(x,y))dxdy}{|R_v|}. \tag{4.16}$$

with $|R_u| = \int_{R_u} dxdy$, $|R_v| = \int_{\Omega\backslash R_u} dxdy$, and for $j = 1, ...m$ different constraints. Then the partial variation of u_j and v_j in Eq.(4.16) w.r.t $\boldsymbol{C}$ is given by

$$\nabla_{\boldsymbol{C}} u_j = \frac{G_j(I) - u_j}{|R_u|}\boldsymbol{N}_u, \;\; \nabla_{\boldsymbol{C}} v_j = -\frac{G_j(I) - v_j}{|R_v|}\boldsymbol{N}_u, \tag{4.17}$$

where $\boldsymbol{N}_u$ denotes the outward unit normal of $\boldsymbol{C}$ (region R_u). Note that the outward unit normal for the boundary of the outer region is $-\boldsymbol{N}_u$.

The contour evolution is found by substituting Eqs.(4.17) into the gradient descent flow Eq.(4.15):

$$\frac{\partial \boldsymbol{C}_u}{\partial t} = f\boldsymbol{N}_u, \quad \text{where } f = \sum_{j=1}^{m}(u_j - v_j)\left(\frac{G_j(I(x,y)) - u_j}{|R_u|} + \frac{G_j(I(x,y)) - v_j}{|R_v|}\right). \tag{4.18}$$

We note that this is a generalized form of the data term of the flow proposed by Yezzi, Tsai, and Willsky [149, 150]. In the above equation, the speed f of the contour along its normal direction, can be directly used in the active polygon evolution equation derived in Eq. (4.6), and restated in Eq. (4.7).

- On the other hand, in Eq. (4.14) we may assign variable weights a_1, and a_2 that depend on the regions. An intuitive choice would be to pick the ratio of the area of each region to the total area of the image domain, say $A = |R_u| + |R_v|$. This choice in practice implies taking into account the number of pixels in each region, and would thus lead to a measure that is normalized with respect to the areas of the regions. Letting $a_1 = \frac{|R_u|}{A}$, $a_2 = \frac{|R_v|}{A}$, we then have to take into account in the derivation of the gradient descent flow, the first variations of the coefficient terms as well:

$$\begin{aligned}\frac{\partial \boldsymbol{C}}{\partial t} = \nabla \widehat{JS}_{a,2} \;\; &= \;\; \sum_{j=1}^{m} \frac{1}{2}\frac{|R_u||R_v|}{A^2}\left(2(u_j - v_j)(\nabla u_j - \nabla v_j)\right)\boldsymbol{N}_u \\ &+ \;\; \frac{1}{2}\frac{|R_v|}{A^2}(u_j - v_j)^2\boldsymbol{N}_u - \frac{1}{2}\frac{|R_u|}{A^2}(u_j - v_j)^2\boldsymbol{N}_u.\end{aligned} \tag{4.19}$$

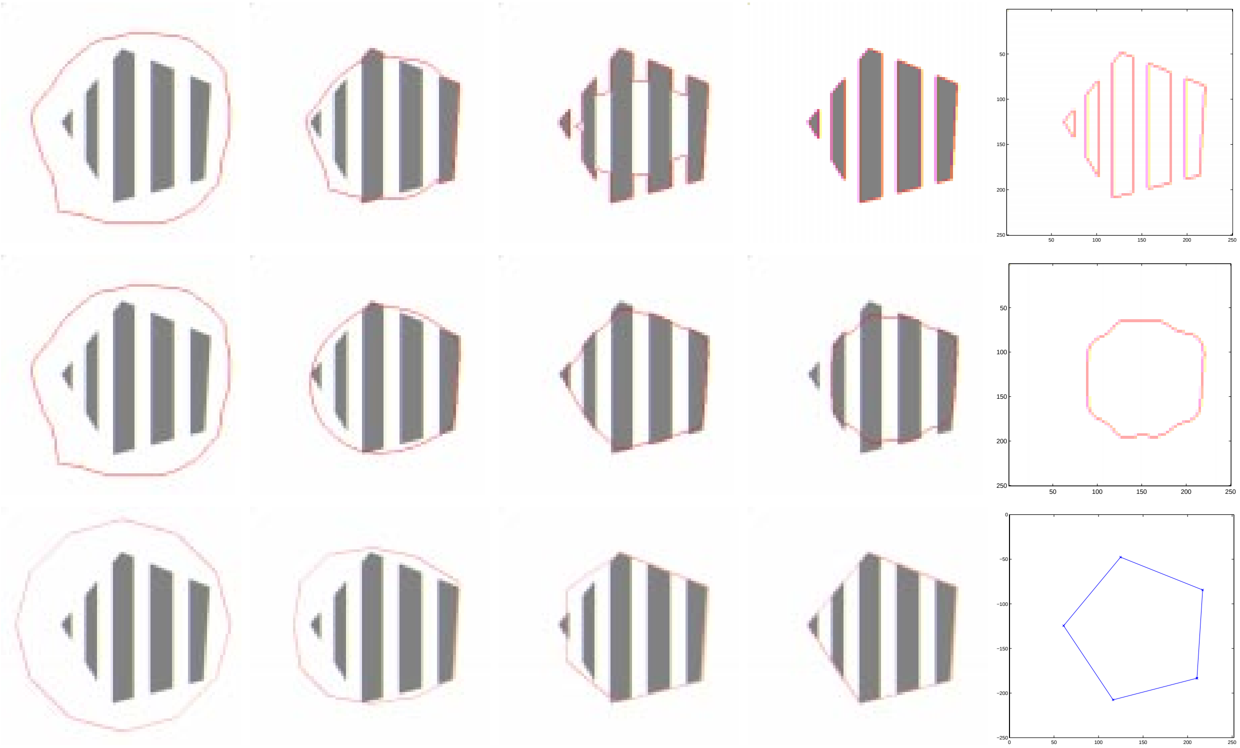


Figure 4.3: A continuous contour with a small regularization (top row), fails to capture as a whole an object with a synthetic texture of vertical stripes , whereas with a large regularization (middle row), rounds off the boundaries, and continues to shrink without sticking to the data. A polygonal contour (bottom row) correctly segments the textured region.

After some manipulations (given in Appendix C), this gradient descent flow can be written as

$$\frac{\partial \boldsymbol{C}}{\partial t} = f\boldsymbol{N}_u, \quad \text{where} \;\; f = \frac{1}{2A}\sum_{j=1}^{m}(u_j - v_j)\left((G_j(I) - u_j) + (G_j(I) - v_j)\right) \tag{4.20}$$

whose energy functional is indeed a generalized form of the external energy functional that Chan and Vese proposed in [29]:

$$E_{CV} = \frac{1}{2A}\sum_{j=1}^{m}\int_{R_u}(G_j(I) - u_j)^2 dxdy + \int_{R_v}(G_j(I) - v_j)^2 dxdy. \tag{4.21}$$

We also note here that in the presence of severe noise, as illustrated in Fig. 4.1 for a uniform noise, the continuous contour ends up encircling small noisy regions, and its length grows. To overcome such effects, a penalty on the length of an active contour is added to its energy functional $E = \iint_R f dxdy + \alpha \oint_{\boldsymbol{C}} ds$, where $\alpha \geq 0$, and s is the arclength of $\boldsymbol{C}$ (also explained in Section 2.3.3). This second term brings a problematic trade-off which we avoid in active polygon

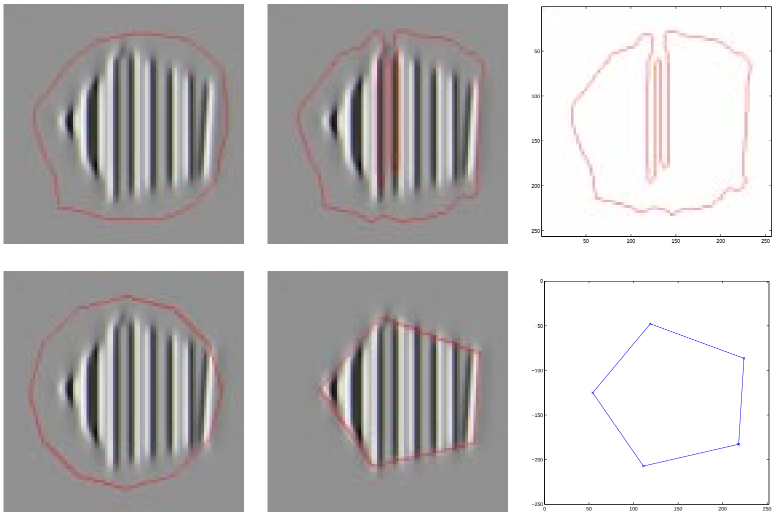


Figure 4.4: An active contour (first row) fails to capture synthetic texture of vertical stripes even after a Gabor filtering, whereas the active polygon (second row) captures the outline of the textured region.

framework as will be explained next.

The advantage of propagating a polygon instead of a continuous curve, is demonstrated in Fig.4.3, where a synthetic texture with vertical stripes is given. Obviously, thresholding for segmentation will not work since one of the stripes has the same color as the background. A continuous active contour propagation (first row of Fig.4.3) fails with a small regularization, i.e. small α mentioned in the previous paragraph, because the curve dips down into the gaps of the stripes with the same color as the background, and the individual bars are captured. With an increased amount of regularization (higher α) on the continuous curve (second row of Fig.4.3), the curve remains intact, however corners are rounded, and if one maintains the evolution, the important features will be “missed”. We have not been successful at obtaining the appropriate tradeoff between the data term and the regularizer in the continuous case to yield a satisfactory result. Active polygon propagation in Eq.(4.6) with f in (4.20) (third row of Fig.4.3), accurately and consistently captures the target shape. We also show in Fig.4.4, the Gabor-filtered version of the same synthetic textured object, (tuned for vertical orientation). It, however, can be observed that the active contour still fails to operate on the filtered texture, whereas the active polygon again successfully captures the outline of the target region.

4.4.2 Evolution of Multiple Active Contours

A nice property of the Jensen-Shannon divergence measure is that it may be generalized as a probabilistic difference measure among any finite number of probability densities. Coupled propagation equations for multiple active contours which delineate different regions on an image domain can also be obtained as the gradient ascent flow of the JS divergence measure among the densities of those regions (note that we are trying to maximize the distance among the densities of regions). There is flexibility in the placement of the contours, which may delineate distinct or overlapping regions without difficulty. We again derive the gradient flows generally with respect to continuous contours. Figure 4.5 depicts two active contours whose inner regions are denoted by R_u, and R_v, and their common exterior by R_w. The measurements, i.e. statistics, in these respective regions are denoted by $u_j, v_j, w_j,\ j = 1, \ldots, m$ for m different measurements, with respective prior probabilities $a_1, a_2, a_3 : a_1 + a_2 + a_3 = 1$. The energy functional for three densities can be written as

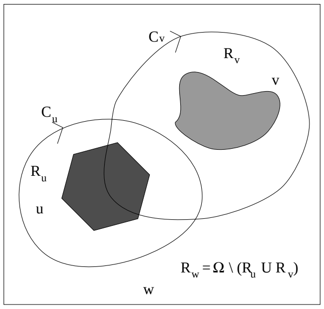


Figure 4.5: Ternary image regions.

$$\begin{aligned}
JS_{a,3} &= H\left(\sum_{i=1}^{3} a_i p_i(\xi)\right) - \sum_{i=1}^{3} a_i H(p_i(\xi)) \\
&\approx H(\nu) - \frac{1}{2}\sum_{j=1}^{m}(a_1\, u_j + a_2\, v_j + a_3\, w_j)^2 \\
&\quad - a_1\left(H(\nu) - \frac{1}{2}\sum_j u_j^2\right) - a_2\left(H(\nu) - \frac{1}{2}\sum_j v_j^2\right) - a_3\left(H(\nu) - \frac{1}{2}\sum_j w_j^2\right) \\
\widehat{JS}_{a,3} &= \frac{1}{2}\sum_{j=1}^{m}\left(a_1\, a_2(u_j - v_j)^2 + a_1\, a_3(u_j - w_j)^2 + a_2\, a_3(v_j - w_j)^2\right) \qquad (4.22)
\end{aligned}$$

This form of energy functional may easily be extrapolated to three active contours, thus four regions with statistics u_j, v_j, w_j, z_j,

$$\widehat{JS}_{a,4} = \frac{1}{2}\sum_{j=1}^{m} \quad a_1\, a_2(u_j - v_j)^2 + a_1\, a_3(u_j - w_j)^2 + a_1\, a_4(u_j - z_j)^2 \\ + a_2\, a_3(v_j - w_j)^2 + a_2\, a_4(v_j - z_j)^2 + a_3\, a_4(w_j - z_j)^2. \qquad (4.23)$$

The prior probabilities of each density $a_1, \ldots, a_4$ may be selected in a variety of ways with the simplest selection being the equal priors, i.e. $a_i = 0.25,\ i = 1, \ldots, 4$. It is now straightforward to approximate the JS functional $\widehat{JS}_{a,N}$ for N regions on the image domain. It may be observed that the first-order approximations to both the densities and the corresponding entropies of the regions, lead to an overall measure that computes a weighted sum of the divergence measures (i.e. distances between their statistics) between all pairwise combinations of the regions.

Active contour evolutions for three regions using means and variances as statistics of each region were derived from a totally different perspective by Yezzi, Tsai, and Willsky [149]. Their energy functional was based on a geometric notion which maximized the area of a triangle formed by the statistics of the three regions. Our energy functional on the other hand, is information-theoretic in nature, and evaluates the distance among probability densities.

The ternary case entails the derivation of a gradient flow for each of the two active contours $\boldsymbol{C}_u$, and $\boldsymbol{C}_v$. Taking the first variation of the energy functional in Eq.(4.22) w.r.t. the contour $\boldsymbol{C}_u$, yields

$$\nabla_{\boldsymbol{C}_u}\widehat{JS}_{a,3} = \sum_{j=1}^{m}[a_1\, a_2(u_j - v_j) + a_1\, a_3(u_j - w_j)]\nabla_{\boldsymbol{C}_u} u_j - [a_1\, a_3(u_j - w_j) + a_2\, a_3(v_j - w_j)]\nabla_{\boldsymbol{C}_u} w_j \qquad (4.24)$$

where we used the fact that $\nabla_{\boldsymbol{C}_u} v_j = 0,\ \forall j$, since v_j are the statistics inside the contour $\boldsymbol{C}_v$ which do not depend on the contour $\boldsymbol{C}_u$.

The partial variation of w_j's requires more attention than that of u_j's and v_j's since the statistic w_j is calculated over the common exterior of both contours whose boundary may not be smooth when the two contours overlap. We exploit a a similar strategy given in [149] to express a statistic in this third region using characteristic functions χ_u, χ_v over the regions R_u, R_v as

$$w_j = \frac{\int_{\Omega\setminus R_u} G_j(I)(1 - \chi_v)dxdy}{\int_{\Omega\setminus R_u}(1 - \chi_v)dxdy}$$

where the denominator may be renamed as $|R_w|$. The variation of w_j w.r.t. $\boldsymbol{C}_u$ can hence be obtained as

$$\nabla_{\boldsymbol{C}_u} w_j = -\frac{G_j(I) - w_j}{|R_w|}(1-\chi_v)\boldsymbol{N}_u. \tag{4.25}$$

The gradient descent flow of the contour $\boldsymbol{C}_u$ which is equal to $\nabla_{\boldsymbol{C}_u}\widehat{JS}_{a,3}$ is thus obtained as $\frac{\partial \boldsymbol{C}_u}{\partial t} = f_u \boldsymbol{N}_u$, where

$$\begin{aligned} f_u &= \sum_{j=1}^{m}[a_1\ a_2(u_j - v_j) + a_1\ a_3(u_j - w_j)]\frac{G_j(I(x,y)) - u_j}{|R_u|} \\ &+ [a_1\ a_3(u_j - w_j) + a_2\ a_3(v_j - w_j)]\frac{G_j(I(x,y)) - w_j}{|R_w|}(1-\chi_v). \end{aligned} \tag{4.26}$$

By similar arguments, one can write $w_j = \frac{\int_{\Omega\setminus R_v} G_j(I)(1-\chi_u)dxdy}{\int_{\Omega\setminus R_v}(1-\chi_u)dxdy}$, where the denominator is equal to $|R_w|$, the variation of w_j w.r.t $\boldsymbol{C}_v$ can be obtained as

$$\nabla_{\boldsymbol{C}_v} w_j = -\frac{G_j(I) - w_j}{|R_w|}(1-\chi_u)\boldsymbol{N}_v. \tag{4.27}$$

Then taking the first variation of the energy functional in Eq.(4.22) w.r.t. the contour $\boldsymbol{C}_v$, and noting that $\nabla_{\boldsymbol{C}_v} u_j = 0,\ \forall j$, leads to the gradient flow of the contour $\boldsymbol{C}_v$, $\frac{\partial \boldsymbol{C}_v}{\partial t} = f_v \boldsymbol{N}_v$, where

$$\begin{aligned} f_v &= \sum_{j=1}^{m}[-a_1\ a_2(u_j - v_j) + a_2\ a_3(v_j - w_j)]\frac{G_j(I(x,y)) - v_j}{|R_v|} \\ &+ [a_1\ a_3(u_j - w_j) + a_2\ a_3(v_j - w_j)]\frac{G_j(I(x,y)) - w_j}{|R_w|}(1-\chi_u). \end{aligned} \tag{4.28}$$

The two speed functions f_u and f_v when inserted into Eq.(4.6) for two separate polygon propagations, $\boldsymbol{C}_u$ and $\boldsymbol{C}_v$, result in ternary polygonal flows.

We illustrate in Figure 4.6, the ternary case for polygon propagations with f_u and f_v given in Eq.(4.26) and Eq.(4.28). Although, the polygonal contours $\boldsymbol{C}_u$ and $\boldsymbol{C}_v$ evolve separately, their motion is coupled through the variables in the evolution equations that depend on each of the three regions. Two contours move in such a way to maximize the approximate Jensen-Shannon divergence among densities of the three regions, namely R_u: inside the contour $\boldsymbol{C}_u$; R_v: inside the contour $\boldsymbol{C}_v$; R_w: the complement of $R_u \bigcup R_v$. Only means are used as the separating statistics, (i.e., $j = 1, G(x) = x$), and the resulting two polygons are shown in Figure 4.6 (right). Hence, the gain is again two-fold: segmentation of the targets, and their description in terms of a handful of vertices are both achieved.

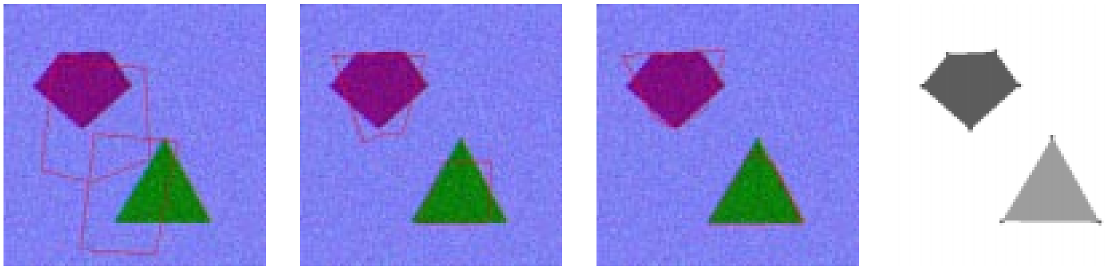


Figure 4.6: Ternary flows using image forces with $G(x) = x$, are used to segment this simple ternary image corrupted by Gaussian noise. Output of the algorithm shown on the right provides only 5 vertex locations for the first object, and 3 vertex locations for the second object.

4.4.3 Active Contours for Vector-Valued Images

Up to now, we have derived the evolutions for grayscale (scalar intensity) images. These results may be extended to vector-valued imagery (e.g. RGB (Red,Green,Blue), or other multi-spectral) with n_{ch} number of different channels. The vector-valued image $\boldsymbol{I} : \mathbb{R}^2 \to \mathbb{R}^{n_{ch}}$, can be expressed in terms of its component functions : $\boldsymbol{I} = (I(x, y, 1), I(x, y, 2), \ldots, I(x, y, n_{ch})$. The two speed functions for the ternary case, may for instance, be computed as

$$\begin{aligned} f_u &= \sum_{l=1}^{n_{ch}} \sum_{j=1}^{m} \{[a_1\ a_2(u_j(l) - v_j(l)) + a_1\ a_3(u_j(l) - w_j(l))] \frac{G_j(I(x,y,l)) - u_j(l)}{|R_u|} \\ &\quad + [a_1\ a_3(u_j(l) - w_j(l)) + a_2\ a_3(v_j(l) - w_j(l))] \frac{G_j(I(x,y,l)) - w_j(l)}{|R_w|}(1 - \chi_v)\}, \\ f_v &= \sum_{l=1}^{n_{ch}} \sum_{j=1}^{m} \{[-a_1\ a_2(u_j(l) - v_j(l)) + a_2\ a_3(v_j(l) - w_j(l))] \frac{G_j(I(x,y,l)) - v_j(l)}{|R_v|} \\ &\quad + [a_1\ a_3(u_j(l) - w_j(l)) + a_2\ a_3(v_j(l) - w_j(l))] \frac{G_j(I(x,y,l)) - w_j(l)}{|R_w|}(1 - \chi_u)\}. \end{aligned} \tag{4.29}$$

4.5 A Global Polygon Regularizer

The flow of an active polygon may, under the sole influence of a data term, become undefined (degenerate) under a variety of scenarios, e.g. when a vertex becomes infinitesimally close to a non-neighbor edge of the polygon, or when two vertices or two edges come infinitesimally close to each other at some point. We show such an example in Fig.4.7, in which propagation of an initial polygon with 5 vertices becomes degenerate, where the vertex $\boldsymbol{V}_4$ approaches (then crosses) the edge between vertices $\boldsymbol{V}_2$ and $\boldsymbol{V}_3$. As a solution to overcome this problem, we introduce a natural regularizing term well adapted to an evolving polygon. Note that the regularizer we are about to

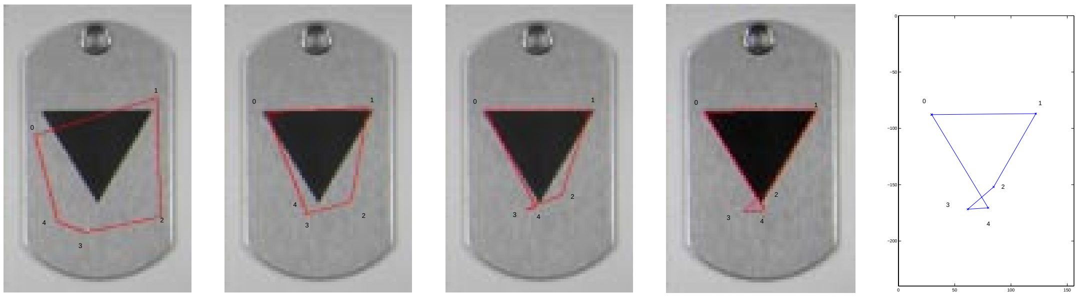


Figure 4.7: We demonstrate that flow (4.6) may become degenerate without an additional constraint on the motion of vertices.

propose is highly adapted to propagating polygons, and would not be computationally feasible for smooth continuous contours. This regularization is accomplished by first viewing each edge of the polygon as a finite line charge of uniform charge density. An electric field $\boldsymbol{E}$, generated by a finite uniform line charge, exerted on a point charge, is directed away from the test charge (we assume all polygon edges have same charge polarity).

4.5.1 Electric force by a line charge:

Given a line charge or a rod of positive charge that extends from a generic point $\boldsymbol{a} \in \mathbb{R}^3$ to $\boldsymbol{b} \in \mathbb{R}^3$, our goal is to calculate the electric field at a point $\boldsymbol{x}' \in \mathbb{R}^3$. Points are chosen in $\mathbb{R}^3$, and using electrostatics principles, we consequently derive these fields for them. The line charge is assumed to be made up of differential point charges dq. We need to compute the differential electric field $d\boldsymbol{E}(\boldsymbol{x}')$ exerted at $\boldsymbol{x}'$ by a charge dq at location $\boldsymbol{x} = \boldsymbol{a} + t(\boldsymbol{b} - \boldsymbol{a})$ which is on the rod (depicted in Figure 4.8). As given by Coulomb's law [138], $d\boldsymbol{E}(x')$ is inversely proportional to the square of the Euclidean distance $||\boldsymbol{x}' - \boldsymbol{x}||^2$ between $\boldsymbol{x}$ and $\boldsymbol{x}'$, and its direction is given by the vector $(\boldsymbol{x}' - \boldsymbol{x})/||\boldsymbol{x}' - \boldsymbol{x}||$, i.e., $d\boldsymbol{E}(\boldsymbol{x}') = (\boldsymbol{x}' - \boldsymbol{x})/||\boldsymbol{x}' - \boldsymbol{x}||^3 dq$. We assume a uniform charge density λ (a constant parameter) along the rod, hence $dq = \lambda dx$. With the change of variable $\boldsymbol{x} = \boldsymbol{a} + t(\boldsymbol{b} - \boldsymbol{a})$, the differential amount of increase in $dx = Ldt$, where $L = ||\boldsymbol{b} - \boldsymbol{a}||$ is the length of the rod extending from $\boldsymbol{a}$ to $\boldsymbol{b}$.

By the principle of superposition, the total electric field $\boldsymbol{E}$ acting on point charge at $\boldsymbol{x}'$ due to line charge $(\boldsymbol{a}, \boldsymbol{b})$, can be obtained by integrating the fields contributed by all differential point

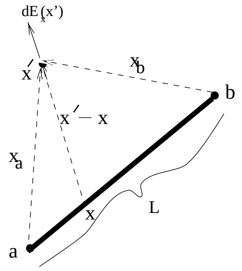


Figure 4.8: Calculation of electric force exerted by a line charge $(\boldsymbol{a}, \boldsymbol{b})$ at a point $\boldsymbol{x}'$ on the polygon.

charges on the rod making up the linear charge distribution,

$$\begin{aligned} \boldsymbol{E}_{ab}(\boldsymbol{x}') &= \int_{\boldsymbol{a}}^{\boldsymbol{b}} \frac{\boldsymbol{x}' - \boldsymbol{x}}{||\boldsymbol{x}' - \boldsymbol{x}||^3} \lambda dx \\ &= k_c \, L \int_0^1 \frac{(\boldsymbol{x}' - \boldsymbol{a}) + t(\boldsymbol{a} - \boldsymbol{b})}{||(\boldsymbol{x}' - \boldsymbol{a}) + t(\boldsymbol{a} - \boldsymbol{b})||^3} dt. \end{aligned} \tag{4.30}$$

Note that the constant λ is combined with the Coulomb's constant into a constant parameter called k_c. After some manipulations (see Appendix D), the total electric field $\boldsymbol{E}_{ab}(\boldsymbol{x}')$ can be written in terms of the two-vectors $\boldsymbol{x}_a = \boldsymbol{x}' - \boldsymbol{a}$ and $\boldsymbol{x}_b = \boldsymbol{x}' - \boldsymbol{b}$ (depicted in Figure 4.8),

$$\boldsymbol{E}_{ab}(\boldsymbol{x}') = \frac{k_c \, L}{||\boldsymbol{x}_a||^2 ||\boldsymbol{x}_b||^2 - (\boldsymbol{x}_a \cdot \boldsymbol{x}_b)^2} \left(\frac{||\boldsymbol{x}_b||^2 \boldsymbol{x}_a - (\boldsymbol{x}_a \cdot \boldsymbol{x}_b)\boldsymbol{x}_b}{||\boldsymbol{x}_b||} + \frac{||\boldsymbol{x}_a||^2 \boldsymbol{x}_b - (\boldsymbol{x}_a \cdot \boldsymbol{x}_b)\boldsymbol{x}_a}{||\boldsymbol{x}_a||} \right). \tag{4.31}$$

As shown in the previous section, a data term corresponding to a given vertex consists of an integrated force along its two neighboring edges. In a similar way, an electric field exerted at a vertex is also integrated along two neighboring edges. Evolution of a polygon vertex $\boldsymbol{V}_k$, due to the total electric field $\boldsymbol{E}_k$ integrated along two neighbor edges $(\boldsymbol{V}_{k-1}, \boldsymbol{V}_k)$ and $(\boldsymbol{V}_k, \boldsymbol{V}_{k+1})$ may then be written as

$$\begin{aligned} \frac{\partial \boldsymbol{V}_k}{\partial t} = \boldsymbol{E}_k &= \sum_{\substack{j=0 \\ j \neq k, j \neq k-1}}^{n} \int_0^1 p \, \boldsymbol{E}_{\boldsymbol{V}_{j-1}, \boldsymbol{V}_j} (p\boldsymbol{V}_k + (1-p)\boldsymbol{V}_{k-1}) \, dp \\ &+ \sum_{\substack{j=0 \\ j \neq k, j \neq k+1}}^{n} \int_0^1 (1-p) \boldsymbol{E}_{\boldsymbol{V}_j, \boldsymbol{V}_{j+1}} ((1-p)\boldsymbol{V}_k + p\boldsymbol{V}_{k+1}) dp. \end{aligned} \tag{4.32}$$

This global, rather than local, geometric dependence makes this regularizer very different from

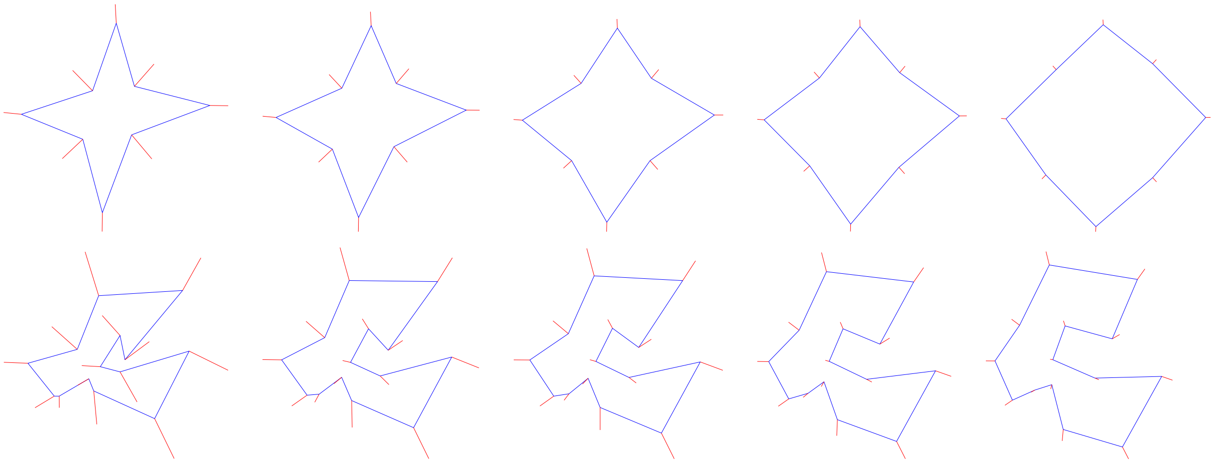


Figure 4.9: Electrostatic regularizer in Eq.(4.32) computes the electric force at each vertex.

the ones used in the literature. The use of a global regularizer is more consistent with the desire to capture local features that would otherwise not be captured. Thus, the additional and novel geometric component of the polygon evolver,which induces global geometric dependence, provides a regularization which both avoids the flow degeneracy as well as captures sharp corners of the target shape without any shrinking or smoothing effects.

We demonstrate the polygon regularization capability of the electric force flow in Eq.(4.32) in Figure 4.9 for two different polygons. The polygonal evolution with an electrostatic regularizer force (plotted at each vertex) is shown at several snapshots in these figures. Note how the force at each vertex is large in magnitude initially, and push a vertex away from the other edges of the polygon. The magnitude of the electric force at each vertex decreases and edges remain well apart which is exactly the effect we expect from the regularizer force. This force should be insignificant when a vertex and its adjacent edges are not very close to most of the other edges, and should become influential, even dominate when the vertex or its adjacent edges are very close to other edges.

The addition of the regularizer term (4.32) to the motion equation of a vertex obtained in Eq.(4.6)

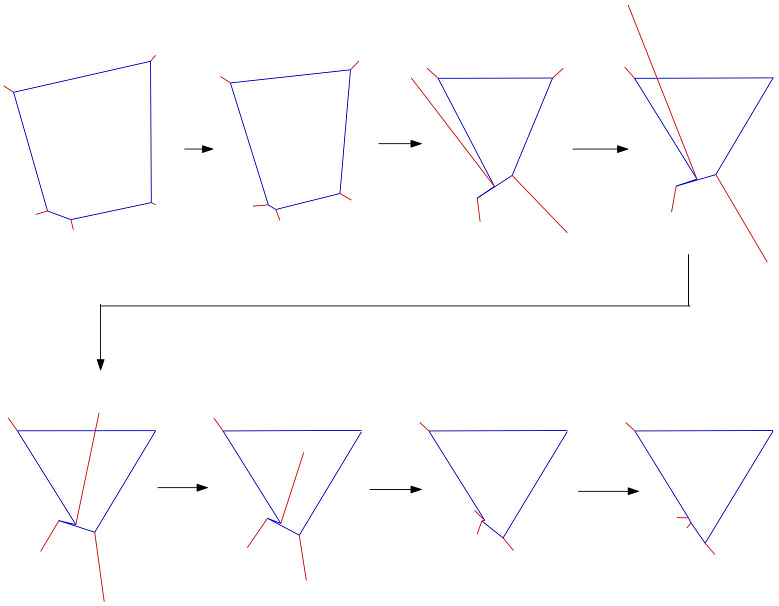


Figure 4.10: The electric forces for an evolution are shown. Note the forces only become significant when a vertex approaches another vertex or edge.

leads to the following modified vertex flow

$$\boxed{\begin{aligned}\frac{\partial \boldsymbol{V}_k}{\partial t} &= \alpha\left\{\int_0^1 pf\big(L(p,\boldsymbol{V}_{k-1},\boldsymbol{V}_k)\big)\,dp\;\boldsymbol{N}_{k,k-1}\right.\\ &\quad+\left.\int_0^1 (1-p)f\big(L(p,\boldsymbol{V}_k,\boldsymbol{V}_{k+1})\big)\,dp\;\boldsymbol{N}_{k+1,k}\right\} + (1-\alpha)\boldsymbol{E}_k. \qquad (4.33)\end{aligned}}$$

Here, α, a constant parameter to weight the influence of data term, and the electric field term, is chosen as 0.95 throughout the evolutions. The reason that we put such a heavy weight on the data term is that the regularizer only kicks in very powerfully when degeneracy occurs, and it lets the data term govern the evolution of the polygon during most of the evolution time.

In Figure 4.10 we demonstrate the use of flow (4.33) for the same initial active polygon with 5 vertices which has been shown to result in an ill-posed flow in the previous section in Figure 4.7. Here, snapshots of only the active polygon on the triangle shape image are given to better appreciate the influence of the regularizer, and we show the electric force at each vertex during this evolution. Note that the electric force at a vertex becomes significantly large when the vertex is infinitesimally close to another edge in this figure. This event exactly keeps the polygon simple during the evolution, and by the effect of the data term, the polygon converges to the target shape.

4.6 Results and Conclusions

In this chapter, we have presented a new approach for image segmentation through polygon propagating equations. In this section, by way of numerical experiments, we validate the effectiveness and the usefulness of our technique.

4.6.1 Implementation Issues

The performance of most active contour algorithms depend on the initial conditions [153]. A particularly important question in carrying out such flows, is that of initializing the contour. Specifically in our case, we need to specify the number of vertices, and their placement to start off the evolution of the polygon. For the active polygons, though, it is possible to circumvent this problem, and speed up the convergence, by a simple approach which helps an initial active polygon adaptively adjust to the number of vertices required for the description of the target shape. Towards that end, we may initialize a very coarse polygon with a small number of vertices, e.g. a big rectangular or circular polygon close to the image boundaries. While the initial polygon, usually with a very few number of vertices, is propagating with both the image force and the regularizer force, new vertices are periodically added and removed affording it a flexible motion towards the target region. A natural criterion to remove a vertex may for instance be based on the angle between its two adjacent edges being close to either 0, or π. This may be effected for a given vertex $\boldsymbol{V}_k$, with its adjacent edges $\boldsymbol{A} = (\boldsymbol{V}_{k-1} - \boldsymbol{V}_k)$, and $\boldsymbol{B} = (\boldsymbol{V}_{k+1} - \boldsymbol{V}_k)$, by computing the following inner product $\boldsymbol{A} \cdot \boldsymbol{B} = ||A|| ||B|| \cos(\theta_{AB})$, the θ_{AB} being the angle between the two vectors. One may therefore check if

$$|\cos(\theta_{AB})| = \frac{|\boldsymbol{A} \cdot \boldsymbol{B}|}{||A|| ||B||} \approx 1, \tag{4.34}$$

to determine the redundancy of a vertex. In either case, the vertex $\boldsymbol{V}_k$ may simply be removed from the vertex list, to finally yield a polygon properly enclosing the shape. During a vertex addition period of an evolution, the magnitude of our image force along each edge of a polygon, i.e. $D = \int_0^1 |f(L(p, \boldsymbol{V}_k, \boldsymbol{V}_{k+1}))| dp$, is computed, and a new vertex is added to a middle point of the edge with the maximum value of D. The intuition here is that the edges with higher image speeds are closer to image structures that may require finer details.

For implementational purposes, the polygon structure P is a two-vector (2-D vector), and the normals needed in the computation of Eq.(4.33) are computed as $\boldsymbol{N}_{k,k-1} = (-P(k,1) + P(k-1,1), \quad P(k,0) - P(k-1,0))^T$. We also maintain a two-dimensional function which acts as an indicator function to show which pixel is inside or outside a polygon during the evolution to facilitate computation of the statistics in the regions. Statistical calculations may be carried out fast by computing the change in the position of each polygon edge, and appropriately adding or subtracting the statistics computed in that difference region, which results in the motion of the edge, which is a straight line, from one iteration to the next. Time step in the discretization of the ODE is chosen large, e.g. $\delta t = 10$, which indeed helps the model to escape various local minima of the objective function. For a typical image of 256×256 size, the relative speed of the active polygon model is approximately twice that of an associated active contour model using the same data term.

As we mentioned, we may also add a random perturbation to our ODE model, a zero-mean Gaussian r.v. with a variance that has a very minor effect when compared to the change in the image term. This trivial perturbation changes the path of each evolution very slightly although the results are very consistent as will be shown next.

4.6.2 Experimental Results

In this section, we demonstrate texture segmentation examples on natural texture images. Our active polygon propagation model is in Eq.(4.33) with speed function in (4.20), and measurement functions $G_1(\xi) = \xi e^{-\xi^2/2}$, $G_2(\xi) = e^{-\xi^2/2}, G_3 = |\xi|, G_4 = \log(cosh(x))$.

In the first example, a zebra on a grassy background constitutes of mainly two textured regions (Fig.4.11). A generic rectangle, i.e. just four vertices, is initialized on the zebra image which is quite challenging in terms of unsupervised texture segmentation. Snapshots from the polygon propagation with the resulting segmentation in Fig.4.11, show that a zebra figure is very nicely captured.

Other natural texture examples include a monarch larvae and a monarch butterfly with generic polygon initializations, a circle and a rectangle, are shown in Fig.4.12. In the same figure, another arbitrary initialization on the same monarch picture, show that the target textured body of the monarch is captured in both cases. In Fig.4.13, a fish whose body has a texture of stripes is captured by an active polygon. Similarly, a sea star on a textured rocky terrain, (Fig.4.14), a cheetah figure (Fig.4.15 left), another cheetah in bushes (Fig.4.15 right), and a chunk of crystal (Fig.4.16) are shown to demonstrate texture capturing capabilities of the active polygon model together with the Jensen-Shannon criterion.

One of our goals was to apply our active polygon model to capture man-made object shapes. A group of real airplane image experiments on each of which a generic circular polygon was initialized, are shown in Fig.4.17. Our model successfully captures the plane shape in terms of a polygon,

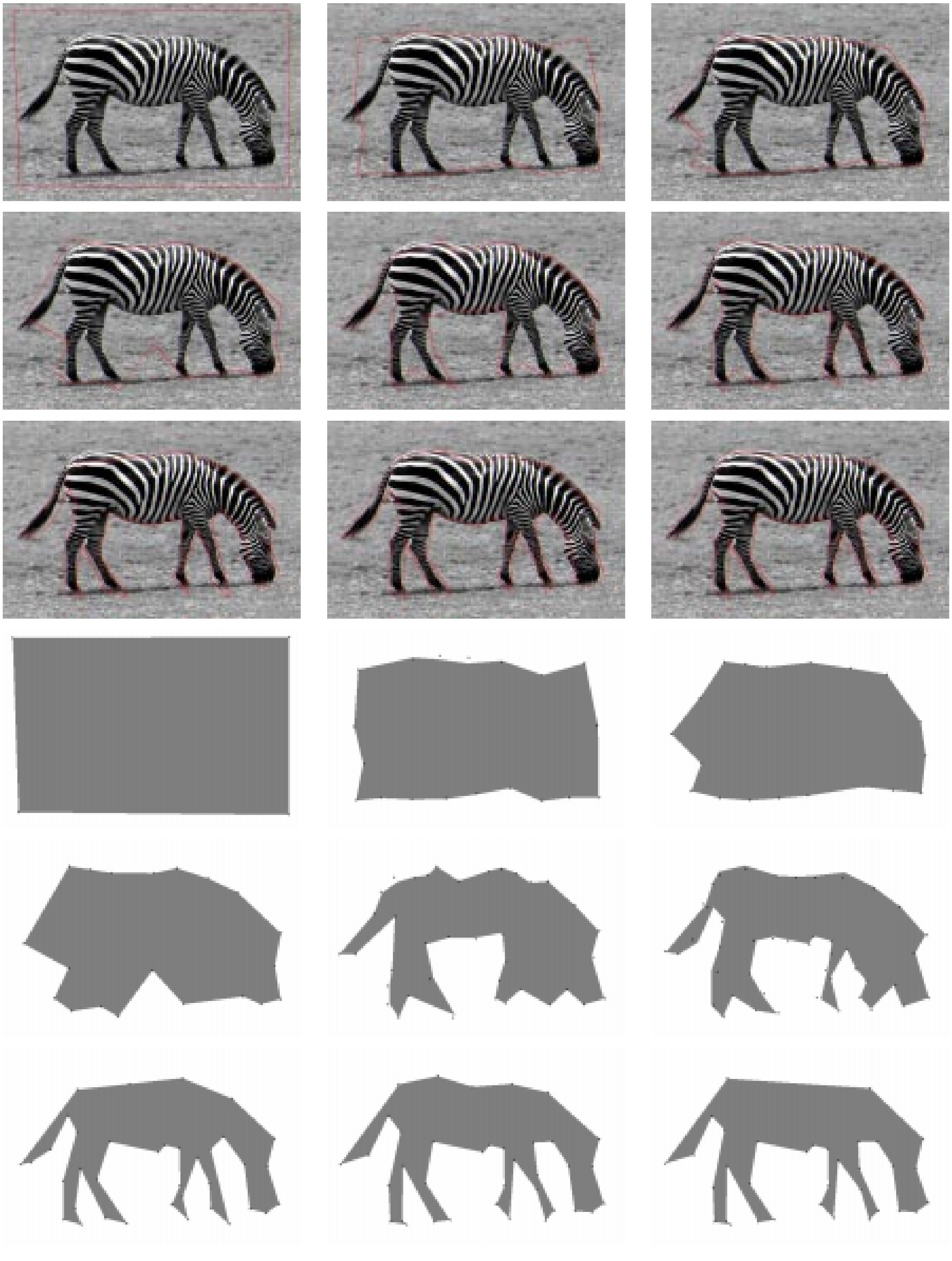


Figure 4.11: A zebra figure is captured by the active polygon model. A generic rectangular active polygon close to image boundaries is initialized.

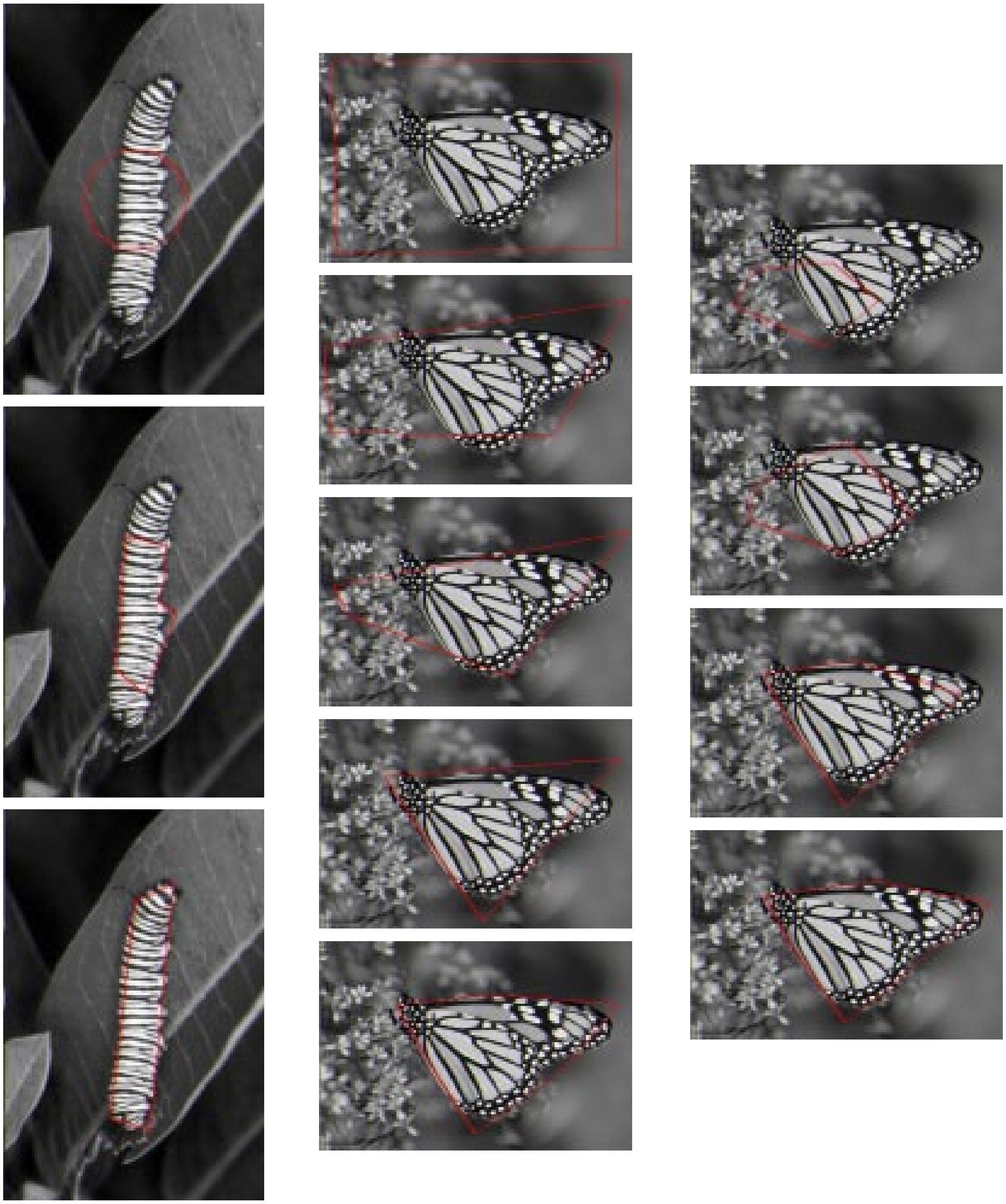


Figure 4.12: A monarch larvae on a leaf is captured by an active polygon (left). Monarch butterfly is captured in the two other columns with very different initalizations.

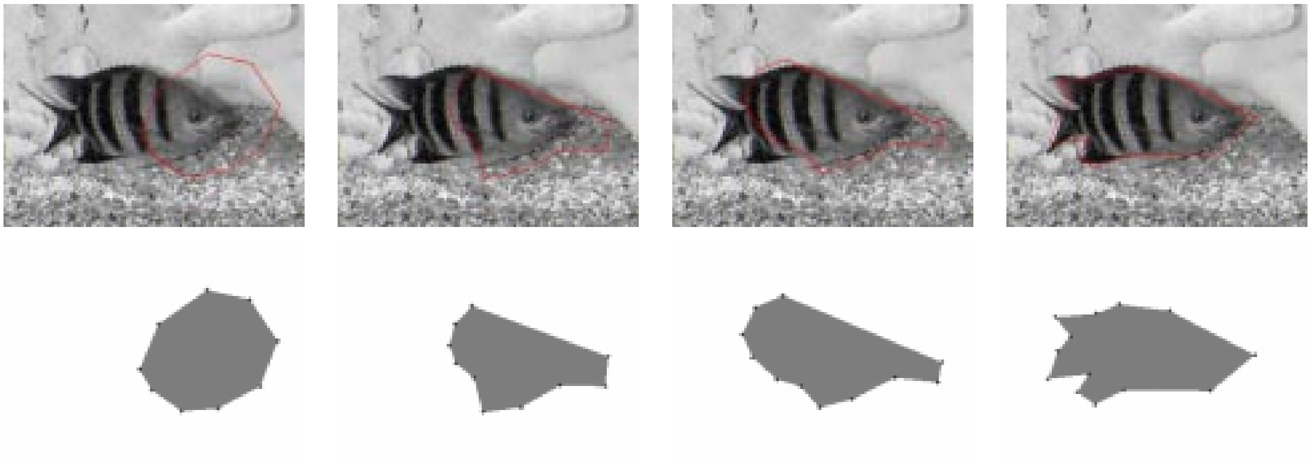


Figure 4.13: A fish with a striped texture is captured.

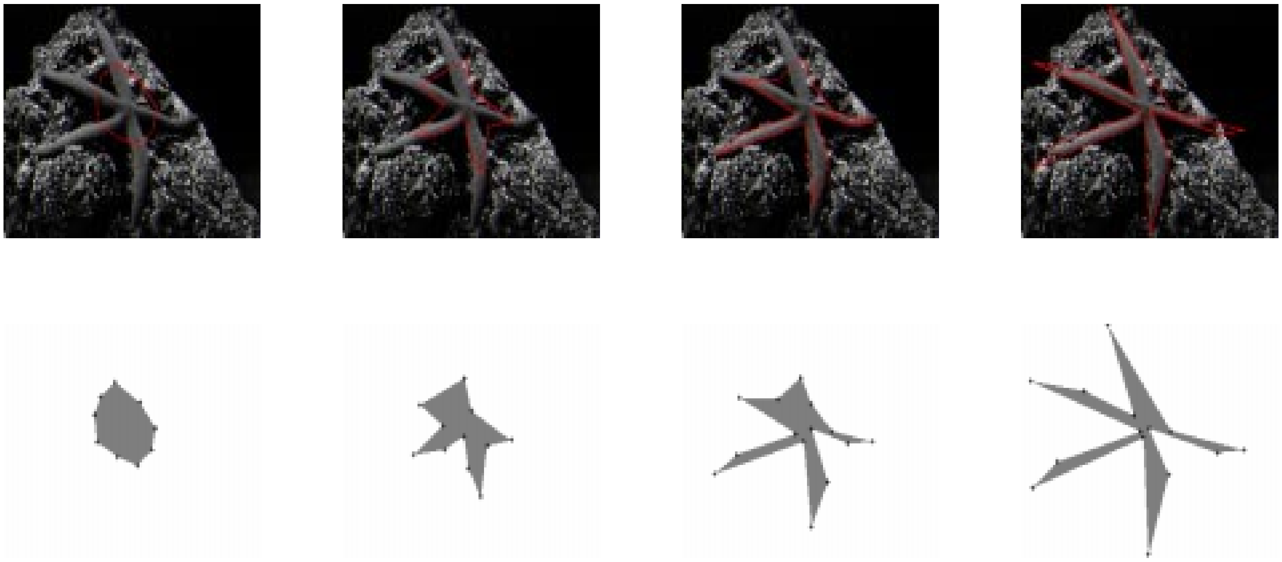


Figure 4.14: A sea star embedded in a textured rocky background is captured.

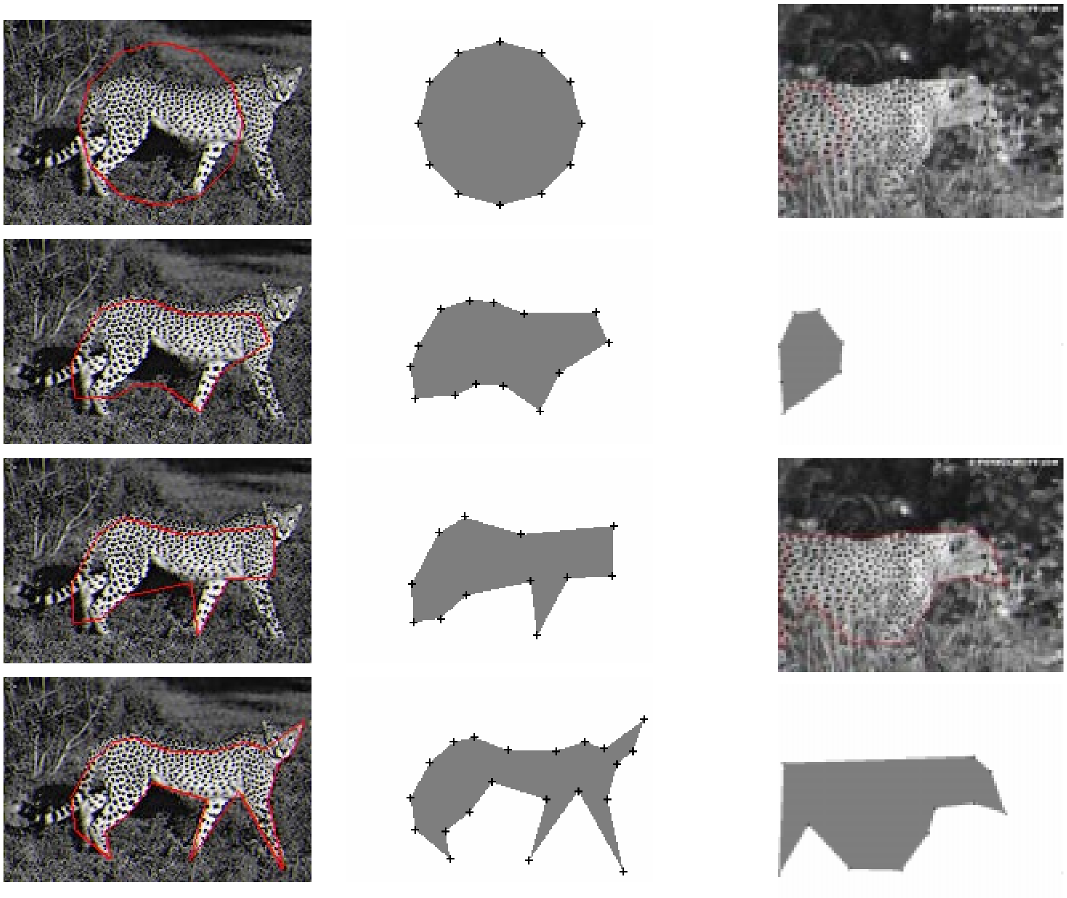


Figure 4.15: Left: Cheetah figure is captured by the active polygon. Right: A cheetah in bushes is captured with an active polygon.

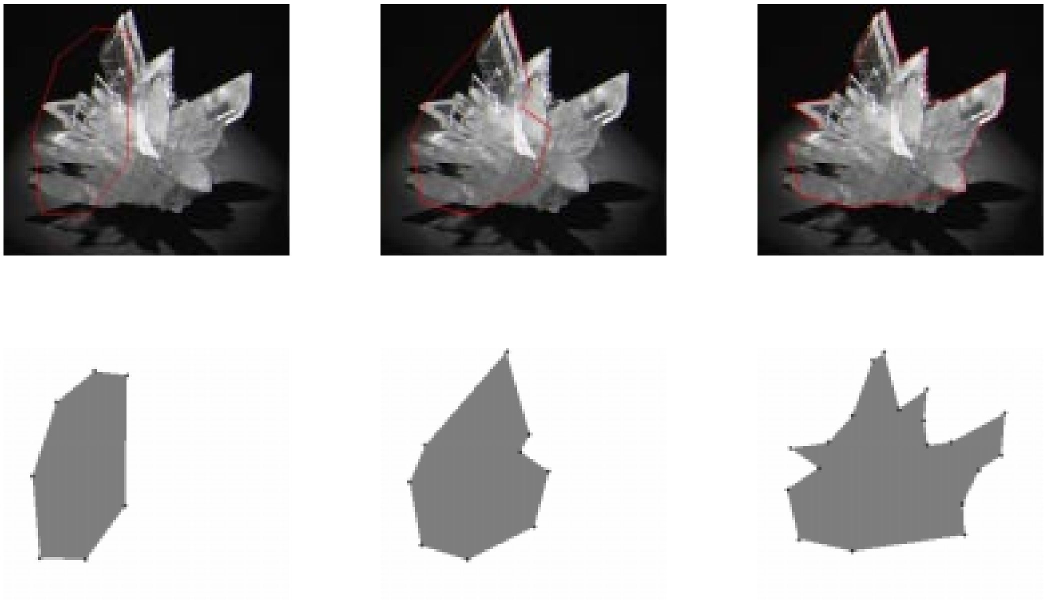


Figure 4.16: A natural crystal chunk is captured.

which may be essential as an input to object recognition algorithms. An image of another man-made object, a submarine, is shown in Fig.4.18.

Our active polygon technique is also suitable for document image segmentation because a document page usually has two different kinds of texture, namely text and images, which have distinct probability distributions. We show two examples of document segmentation: Fig.4.19 with a single active polygon, and Fig.4.20 with two active polygons.

4.6.3 Conclusions

In this chapter, we have presented a polygon propagation model to capture particularly textured objects in images. A new ODE model was developed to move polygon vertices. In addition, a new global polygon regularizer was introduced to avoid degeneracy during polygon propagation. Adaptation of a favorable divergence measure, the Jensen-Shannon divergence, as an integral form (energy functional) of our ODEs lead to quite a powerful unsupervised texture segmentation technique which was validated by numerical results. The only assumption, which is valid in many applications, in our model on the target image regions is, that they should be simply connected. This is on account of most of the cases, whether in natural images as in zebra, or man-made object images as in airplanes, texture regions are simply connected.

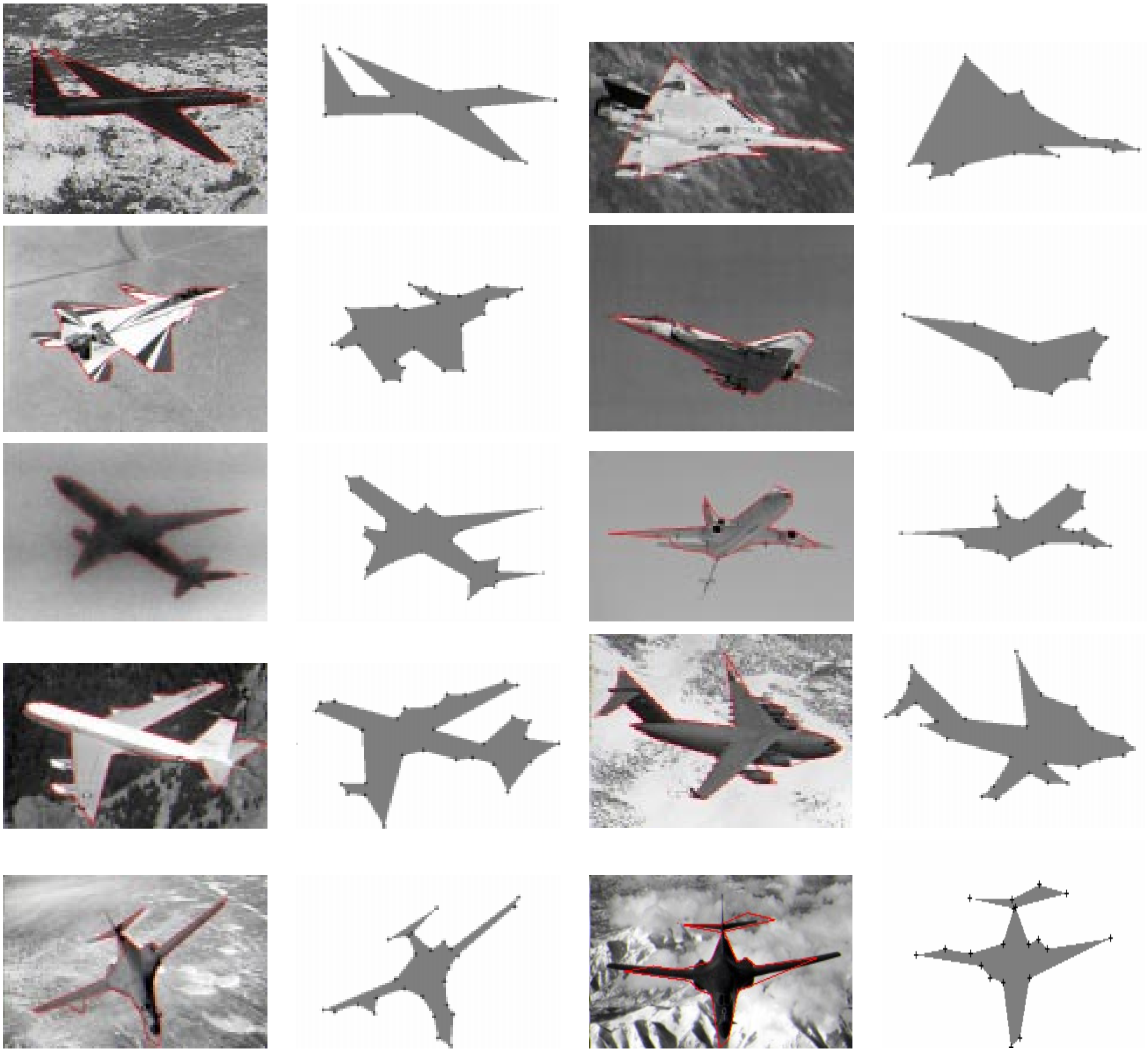


Figure 4.17: Each airplane is captured by a handful of vertices.

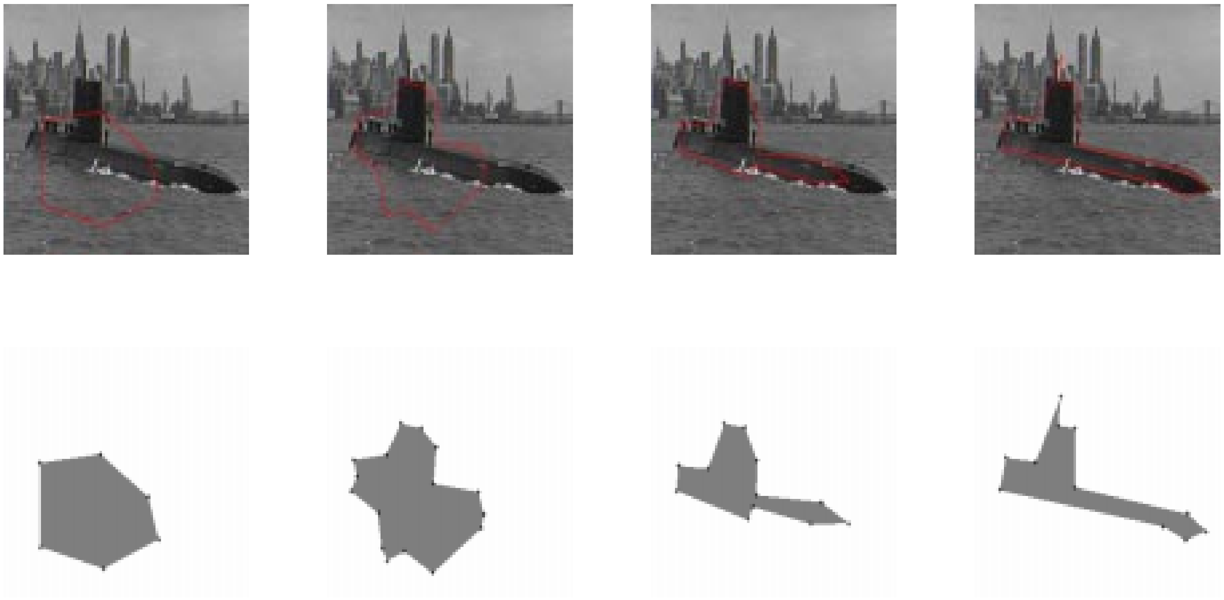


Figure 4.18: A submarine figure is captured by a handful of vertices.

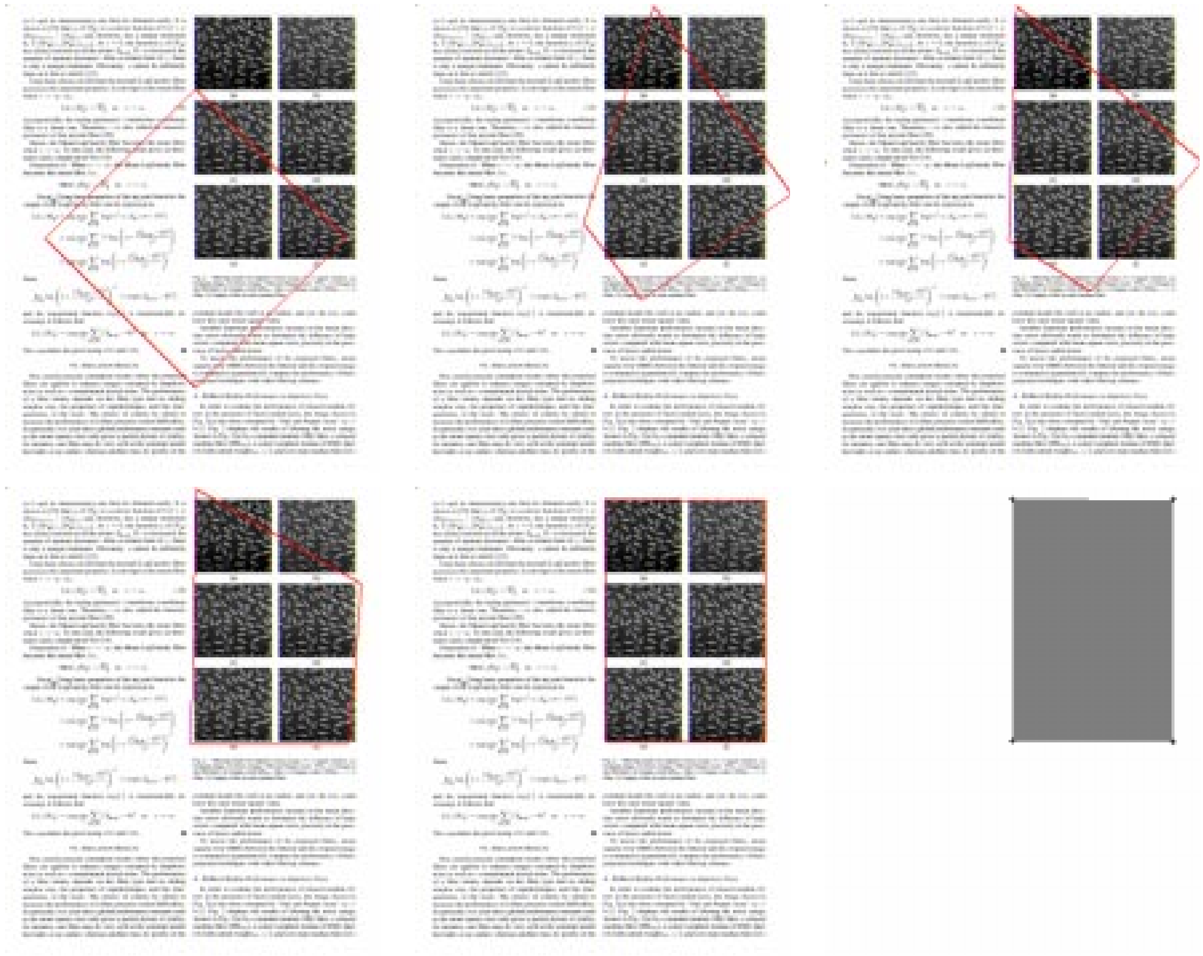


Figure 4.19: An active polygon nicely segments a document image scanned from an article.

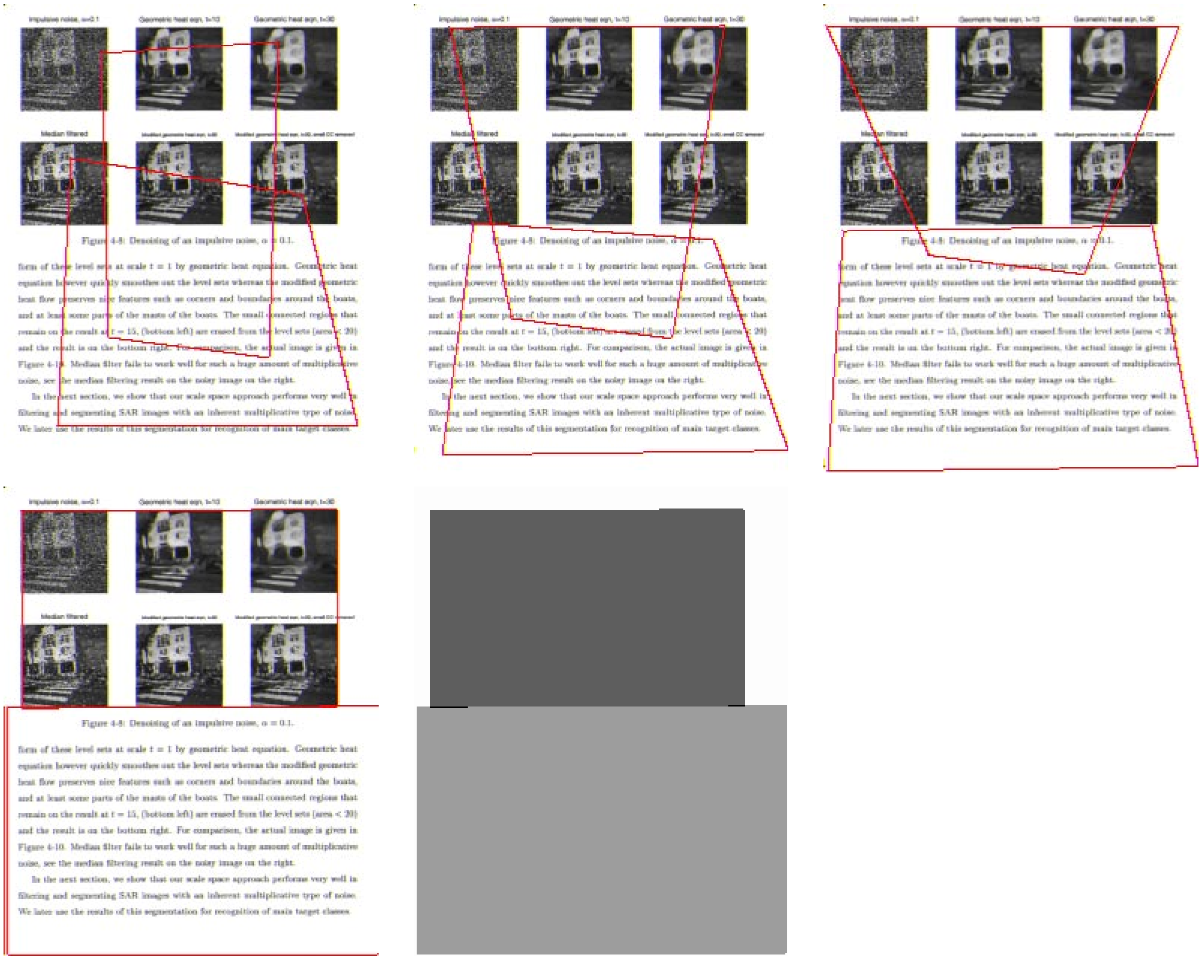


Figure 4-9: Denoising of an impulsive noise, α = 0.1.

form of these level sets at scale t = 1 by geometric heat equation. Geometric heat equation however quickly smoothes out the level sets whereas the modified geometric heat flow preserves nice features such as corners and boundaries around the boats, and at least some parts of the masts of the boats. The small connected regions that remain on the result at t = 15, (bottom left) are erased from the level sets (area < 20) and the result is on the bottom right. For comparison, the actual image is given in Figure 4-10. Median filter fails to work well for such a huge amount of multiplicative noise, see the median filtering result on the noisy image on the right.

In the next section, we show that our scale space approach performs very well in filtering and segmenting SAR images with an inherent multiplicative type of noise. We later use the results of this segmentation for recognition of main target classes.

Figure 4.20: Two active polygons capture text and image regions of a page from an article.

Chapter 5

Applications of Active Polygons

Often, the goal of *low level vision* is to extract significant features, e.g. boundaries of objects, from an image, for a subsequent use in recognition by the *high level vision* stage. In this chapter, we present applications tailored to the active polygon model that we have developed for extracting compact descriptions of objects in an image. This representation is shown to be useful in higher level applications. With a resulting handful of vertices for an object description, our method is highly flexible, and may be efficiently applied to *object tracking* in video sequences, and *object recognition* in visual information retrieval context:

1. **Video Object Tracking:** We present a technique for object tracking in image sequences, which makes the use of the active polygon framework developed in Chapter 4. In this approach, upon capturing boundaries of an object by a polygon from the first few frames of the image sequence, motion segmentation of a polygonal object can be quickly achieved by invoking only the vertex locations and the adjacent edges in the formulation. We carry out a velocity field estimation at an active polygon vertex using the optical flow constraint on its two adjacent edges. A spatial segmentation phase follows to further refine the object's vertex locations estimated by the optical flow. The advantage of our region-based active polygons over continuous active contours in object tracking in video applications is highlighted by its provision of a compact representation of object features, particularly for simply connected target shapes, hence will be essential for their tracking [143].

2. **Object Recognition:** We present a technique for object recognition which makes use of the compact shape representation extracted from an image by the active polygons. A signature function computed from the polygonal representation is matched with the signatures of the

model objects in a small library, via a cross-correlation operation. An intra-class retrieval experiment, using photographs of model airplanes is carried out, and it is shown that active polygons can be successfully applied to such problems.

These two applications are explored in the remainder of this Chapter.

5.1 Active Polygons for Object Tracking

5.1.1 Overview on Video Object Tracking Methods

Recent developments in digital technology have increased acquisition of digital video data, which in turn have led to more applications in video processing. Video sequences provide additional information about how scenes and objects change over time when compared to still images. The problem of tracking moving objects remains of great research interest in computer vision on account of various applications in video surveillance, monitoring, robotics, and video coding. For instance, MPEG-4 video standard introduced video object plane concept, and a decomposition of sequences into object planes with different motion parameters [73]. Video surveillance systems are needed in traffic and highway monitoring, in law enforcement and security applications by banks, stores, and parking lots. Algorithms for extracting and tracking over time moving objects in a video sequence are hence of importance.

Tracking methods may be classified into two categories [114]: (i) **Motion-based approaches**, which use motion segmentation of temporal image sequences by grouping moving regions over time, and by estimating their motion models [10, 19, 99, 145]. This region tracking, not being object-based is not well-adapted to the cases where a prior shape knowledge of the moving object is provided. (ii) **Model-based approaches** track objects using a template of the 3D object such as 3D models in [50, 79, 91, 95, 120]. Usage of this high level semantic information, yields robust algorithms at a high computational complexity cost.

Another classification of object tracking methods due to [114] is based on the type of information that the tracking algorithm uses. Along these lines, tracking methods which exploit either boundary-based information or region-based information, have been proposed:

1. **Boundary-based methods** use the boundary information along the object's contour, and are flexible because usually no object shape model and no motion model are required. However, methods using snake models such as [12, 49, 67], employ parameterized snakes (such as B-

splines), and constrain the motion by assuming certain motion models, e.g. rigid, or affine. In [12], a contour's placement in a subsequent frame is predicted by an iterative registration process, where rigid objects, and rigid motion are assumed. In another tracking method with snakes [89], the motion estimation step is skipped, and the snake position from any given image frame is carried to the next frame. Other methods employ geodesic active contour models [24] (which also assumes rigid motion and rigid objects), and [114].

2. **Region-based methods** such as [10, 99, 145] segment a temporal image sequence into regions with different motions. Regions segmented from each frame by a motion segmentation technique are matched to estimate motion parameters [9]. They usually employ parametric motion models, and they are computationally more demanding than boundary-based tracking methods because of the cost of matching regions.

Another tracking method, referred to as Geodesic Active Regions [113], incorporates both boundary-based and region-based approaches. An affine motion model is assumed in this technique, and its computational cost is significant on account of many different estimation steps involved.

Our goal is to build on the existing achievements, and the corresponding insight to develop a simple and efficient boundary-based tracking algorithm well adapted to polygonal objects. This is in effect an extension of our evolution models which use region-based data distributions to capture polygonal object boundaries .

5.1.2 Motion Estimation

Motion of objects in 3D real world scene are projected onto 2D image plane, and this projected motion is referred to as "apparent motion", or "2D image motion", or sometimes as "optical flow", which is to be estimated. In a time-varying image sequence, $I(x, y, t) : [0, a] \times [0, b] \times [0, T] \to \mathbb{R}^+$, image motion may be described by a 2-D vector field $\boldsymbol{V}(x, y, t)$, which specifies the direction and speed of the moving target at each point (x, y). The measurement of visual motion is equivalent to computing $\boldsymbol{V}(x, y, t)$ from $I(x, y, t)$ [60]. Estimating the velocity field remains an important research topic in light of its ubiquitous presence in many applications and as reflected by the wealth of previously proposed techniques.

The most popular group of motion estimation techniques are referred to as **differential techniques**, and solve an optical flow equation which states that intensity or brightness of an image remains constant with time. They use spatial and temporal derivatives of the image sequence in a gradient search, and sometimes referred to as **gradient-based techniques**. The basic assumption that a point in the 3D shape, when projected onto the 2D image plane, has a constant intensity over time, may be formulated as ($\boldsymbol{x} = (x, y)$),

$$I(\boldsymbol{x}, t) \approx I(\boldsymbol{x} + \delta\boldsymbol{x}, t + \delta t)$$

where $\delta\boldsymbol{x}$ is the displacement of the local image region at $(\boldsymbol{x}, t)$ after time δt. A first-order Taylor series expansion on the right-hand-side yields

$$I(\boldsymbol{x}, t) = I(\boldsymbol{x}, t) + \nabla I \cdot \delta\boldsymbol{x} + I_t \delta t + O^2$$

where O^2 denotes second and higher order derivatives. Dividing both sides of the equation by δt, and neglecting O^2, the **optical flow constraint equation** is obtained as

$$\frac{\partial I}{\partial t} + \nabla I \cdot \boldsymbol{V} = 0. \tag{5.1}$$

This constraint is, however, not sufficient to solve for both components of $\boldsymbol{V}(\boldsymbol{x}, t) = (u(\boldsymbol{x}, t), v(\boldsymbol{x}, t))$, and additional constraints on the velocity field are required to address the ill-posed nature of the problem.

The direction of motion of an object boundary B monitored through a small aperture A (small with respect to the moving unit) (see Figure 5.1) can not be determined uniquely (known as the **aperture problem**). Experimentally, it can be observed that when viewing the moving edge B through aperture A, it is not possible to determine whether the edge has moved towards the direction c or direction d. The observation of the moving edge only allows for the detection and hence computation of the velocity component normal to the edge (vector towards n in Figure 5.1), with the tangential component remaining undetectable. Uniquely determining the velocity field hence requires more than a single measurement, and it necessitates a combination stage using the local measurements [139]. This in turn means that computing the velocity field involves regularizing constraints such as its smoothness and other variants.

Horn and Schunck, in their pioneering work [63], combined the optical flow constraint with a global smoothness constraint on the velocity field to define an energy functional whose minimization

$$\arg\min_{u,v} \int_{\Omega} [(\nabla I \cdot \boldsymbol{V} + I_t)^2 + \lambda^2(||\nabla u||^2 + ||\nabla v||^2)] d\boldsymbol{x},$$

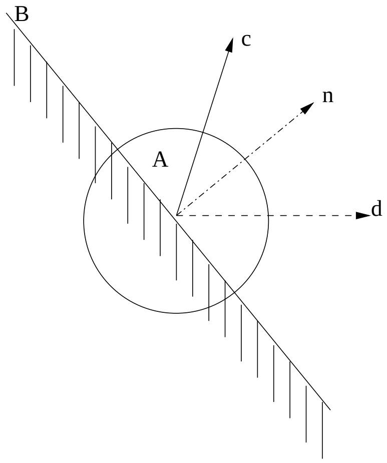


Figure 5.1: The aperture problem: when viewing the moving edge B through aperture A, it is not possible to determine whether the edge has moved towards the direction c or direction d.

can be carried out by solving its gradient descent equations. A variation on this theme, would adopt an L_1 norm smoothness constraint, (in contrast to Horn-Schunck's L_2 norm), on the velocity components, and was given in [82]. Lucas and Kanade, in contrast to Horn and Schunck's regularization based on post-smoothing, minimized a pre-smoothed optical constraint

$$\int_R W^2(\boldsymbol{x})[\nabla I(\boldsymbol{x},t) \cdot V + I_t(\boldsymbol{x},t)]^2,$$

where $W(\boldsymbol{x})$ denotes a window function that gives more weight to constraints near the center of the neighborhood R [92].

Imposing the regularizing smoothness constraint on the velocity over the whole image leads to oversmoothed motion estimates at the discontinuity regions such as occlusion boundaries and edges. Attempts to reduce the smoothing effects along steep edge gradients included modifications such as incorporation of an oriented smoothness constraint by [105], or a directional smoothness constraint in a multiresolution framework by [47]. Hildreth [60] proposed imposing the smoothness constraint on the velocity field only along contours extracted from time-varying images. One advantage of imposing smoothness constraint on the velocity field is that it allows for the analysis of general classes of motion, i.e., it can account for the projected motion of 3D objects that move freely in space, and deform over time [60].

Spatio-temporal energy-based methods make use of energy concentration in 3D spatio-temporal frequency domain. A translating 2D image pattern transformed to the Fourier domain shows that its velocity is a function of its spatio-temporal frequency [11]. A family of Gabor filters which simultaneously provide spatio-temporal and frequency localization, were used to estimate velocity components from the image sequences [46, 59].

Correlation-based methods estimate motion by correlating or by matching features such as edges, or blocks of pixels between two consecutive frames [135], either as block matching in spatial domain, or phase correlation in the frequency domain. Similarly, in another classification of motion estimation techniques, **token-matching schemes**, first identify features such as edges, lines, blobs or regions, and then measure motion by matching these features over time, and detecting their changing positions [139]. There are also **model-based approaches** to motion estimation, and they use certain motion models. Much work has been done in motion estimation, and the interested reader is referred to [11, 128, 133, 135] for a more compulsive literature.

5.1.3 Our Approach

The key idea in our proposed technique for video sequences is centered around tracking a relatively few vertices together with their corresponding edges, which in turn yields a bookkeeping simplicity and hence efficiency. For a specifically fast object tracking goal, it may be sufficient to only measure certain properties of the vector field $\boldsymbol{V}(x, y, t)$, as a more complete and precise characterization is rather redundant. Moreover, a very sparse set of motion measurements only along the contour points, is required in lieu of a dense velocity field estimation over the whole image. A velocity estimation is proposed in tandem with a fast spatial and motion segmentation which follow the initial object delineation in the first few frames. We carry out a velocity field estimation at an active polygon vertex using the optical flow constraint on its two adjacent edges. A spatial segmentation phase follows for a further refinement of the object's vertex locations estimated by the optical flow. As discussed below, this approach presents several advantages over active contours in video tracking; not the least of which is its feature parsimony which in turn, greatly simplifies any tracking application.

Inspired by our spatial image segmentation on a still image, we achieve an incorporation of a velocity field estimation in case of an image sequence as described next.

5.1.4 Velocity Estimation At Vertices

Recall that our polygon propagating ODEs written for each vertex $\boldsymbol{P}_k$ laid on a single image are given by

$$\begin{aligned} \frac{\partial \boldsymbol{P}_k}{\partial t'} &= \boldsymbol{u}^{\perp}_{1,k} \int_0^1 pf\big(L(p, \boldsymbol{P}_{k-1}, \boldsymbol{P}_k)\big)\, dp \\ &+ \boldsymbol{u}^{\perp}_{2,k} \int_0^1 (1-p) f\big(L(p, \boldsymbol{P}_k, \boldsymbol{P}_{k+1})\big)\, dp\ , \end{aligned} \tag{5.2}$$

where $\boldsymbol{u}^{\perp}_{1,k}$ (resp. $\boldsymbol{u}^{\perp}_{2,k}$) denotes the outward unit normal of edge $(\boldsymbol{P}_{k-1} - \boldsymbol{P}_k)$ (resp. $(\boldsymbol{P}_k - \boldsymbol{P}_{k+1})$), and t' denotes the evolution time for the differential equation. Our polygonal flows essentially integrate spatial image information along two adjacent edges of a vertex $\boldsymbol{P}_k$ to determine its speed and direction. This is exactly the same idea we will use in the estimation of a velocity field at a vertex of the active polygon laid on a time-varying image sequence. Our goal is to estimate the velocity vector at each vertex $\boldsymbol{P}_k$ using the two adjacent edges as shown in Fig. 5.2. The velocity field $\boldsymbol{V}(x,y)$ at each point on an edge may be represented as $\boldsymbol{V}(p) = v^{\perp}(p)\boldsymbol{u}^{\perp}(p) + v^T(p)\boldsymbol{u}^T(p)$, where $\boldsymbol{u}^T(p)$ and $\boldsymbol{u}^{\perp}(p)$ are unit vectors in the directions tangent and normal to an edge. The optical flow constraint given by

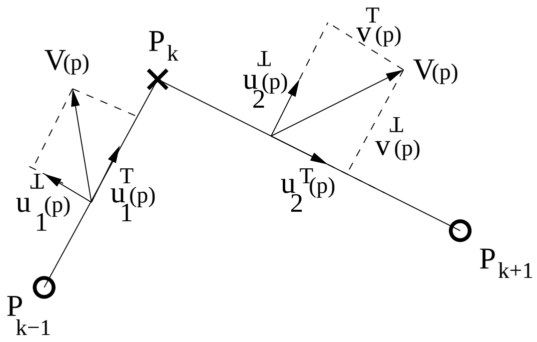


Figure 5.2: 2-D velocity field along two neighbor edges of a polygon vertex.

$$\frac{\partial I}{\partial t} + \nabla I \cdot \boldsymbol{V} = 0, \tag{5.3}$$

provides a way to estimate the component $v^{\perp}$ of the velocity field directly from the time-varying image $I(x,y,t)$. Once an active polygon locks onto a target polygonal object, the unit direction vectors $\boldsymbol{u}^{\perp}$ and $\boldsymbol{u}^T$ are also known immediately. A set of local measurements may easily be obtained from the image brightness constraint

$$v^{\perp}(x,y) = -I_t(x,y)/||\nabla I(x,y)||, \tag{5.4}$$

which gives the magnitude of velocity field in the direction orthogonal to the local edge structure. This initial pointwise measurement is preferred, because different parts of the moving object may have different motions under general conditions such as unrestricted motion [139]. However, instantaneous measurements are insufficient to determine the motion, and the local measurements must somehow be integrated for a better estimation of the velocity at a point. Recall that this was also motivated by the aperture problem which showed the necessity of a combination stage over several measurements on the image. For a fast estimation of a velocity at a polygon vertex, we utilize a joint contribution from two edges of a vertex to infer the resultant motion at the vertex. Expecting a sensitivity of such instantaneous normal velocity measurements to noise, we integrate them in a weighted manner along two neighboring edges of a vertex to yield a more robust estimation. This leads us to write a velocity estimation scheme at each vertex of an active polygon as

$$\begin{aligned} \frac{\partial \boldsymbol{P}_k}{\partial t} = \boldsymbol{V}_k &= \boldsymbol{u}_{1,k}^{\perp} \frac{\int_0^1 p \, v^{\perp}\big(L(p, \boldsymbol{P}_{k-1}, \boldsymbol{P}_k)\big) dp}{\int_0^1 p \; dp} \\ &+ \boldsymbol{u}_{2,k}^{\perp} \frac{\int_0^1 (1-p) \, v^{\perp}\big(L(p, \boldsymbol{P}_k, \boldsymbol{P}_{k+1})\big) dp}{\int_0^1 (1-p) \; dp} \end{aligned} \tag{5.5}$$

for $k = 1, ...n$. To introduce further robustness and to achieve more reliable estimates in the course of computing $v^{\perp}$, we may make use of smoother spatial derivatives (larger neighborhoods) [43, 62, 82].

To proceed with tracking of a polygonal object, we first apply the previously described segmentation technique to initially delineate the moving object boundaries. Upon initialization, we assume that an object of interest has been already outlined. Using Eq. (5.5), the velocity vector is estimated from two images $I(x, y, t)$ and $I(x, y, t+1)$, and is subsequently utilized to move the active polygon's handful of vertices to new locations on the next image $I(x, y, t+1)$ in the sequence. The ODE in Eq. (5.2) is then run for a short time for further refinement of the polygon delineation of the moving object. We substantiate our approach by the tracking examples given in the next section.

5.1.5 Experimental Results

Object Tracking in IR (Infra Red) Video Sequences: A very suitable application of our object tracking method is to target tracking in IR video sequences. In an IR image, a target usually appears as a slightly bright or dark spot, most of the time in the form of a rough polygonal shape, when compared to the background terrain. Our active polygons hence are perfectly suited to locking onto such targets and tracking them by the fast method we have introduced in the previous section.

We show in Fig. 5.3, and Fig. 5.4, several snapshots (progression from-left-to-right, top-bottom in all figures) from two IR sequences respectively, in which the targets, a bright triangular shape in Fig. 5.3, and a bright rectangular shape in Fig. 5.4, are successfully tracked by our technique. The active polygon at each snapshot is depicted in yellow (white in b/w print-outs). Similarly, in Fig. 5.5, a target which is probably a tank, is tracked as can be observed in the given snapshots, and the final representation provided by the active polygon closely resembles a tank-like structure.

Here, it should be emphasized that the gain of our approach is again two-fold because in addition to tracking, the shape of the tracked object is also available with a very compact description, as a bonus feature.

In the next IR sequence example (Fig. 5.6), the tracker approaches gradually to a possible target site, which appears as a bright region, and the polygon adapts to the outline of this target region during tracking. As a last example to IR sequence tracking, Fig. 5.7 shows a quite noisy, cluttered terrain, where again a target site appears as a dark spotted foreground region, highly embedded in the background terrain clutter. The active polygon finds the roughly triangular outline of this site during its successful tracking through the IR image sequence.

In the next example (Fig. 5.8), we tracked a toy airplane, a model of an F22, which is translated on a conveyor belt. Note that once the active polygon locks onto the target object, it follows the movement of the target unaffected by the camera movements such as zoom, pan, translation, and rotation. Similarly, a model rocket which is about to be launched, is zoomed out by the camera in Fig. 5.9, and is well tracked by our algorithm.

The last two tracking examples show a potential application of our technique to traffic monitoring where overhead cameras can give information on state of traffic, its density, the speed of the vehicles, or abnormal situations on the road. In the first sequence, shown in Fig. 5.10, a fixed camera has been mounted on a hilltop overviewing the Tryon Road, Raleigh, NC. A black automobile entering the scene from the left, is successfully tracked by an active polygon until it leaves the scene at the other end of the road hence out of the camera range. In another road sequence (of COST 211 project [1]), shown in Fig. 5.11, a truck approaching from the horizon on the road (left), is tracked via an active polygon. Active polygons are successfully applied to tracking of motor vehicles, because the shapes of these man-made vehicles are polygonal, making the motion tracking through vertices of a polygonal shape feasible and practical.

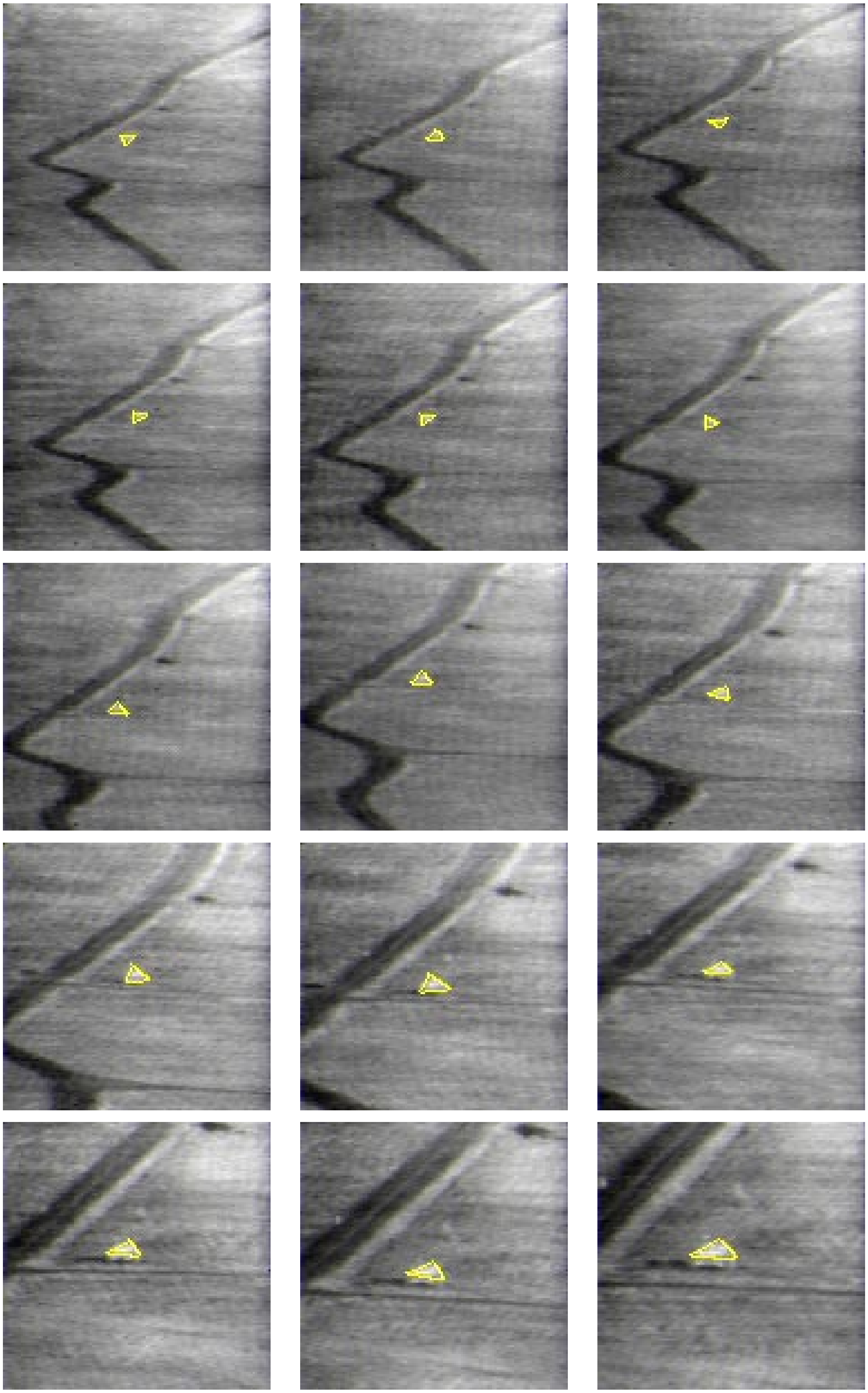


Figure 5.3: Tracking of an IR image sequence I.

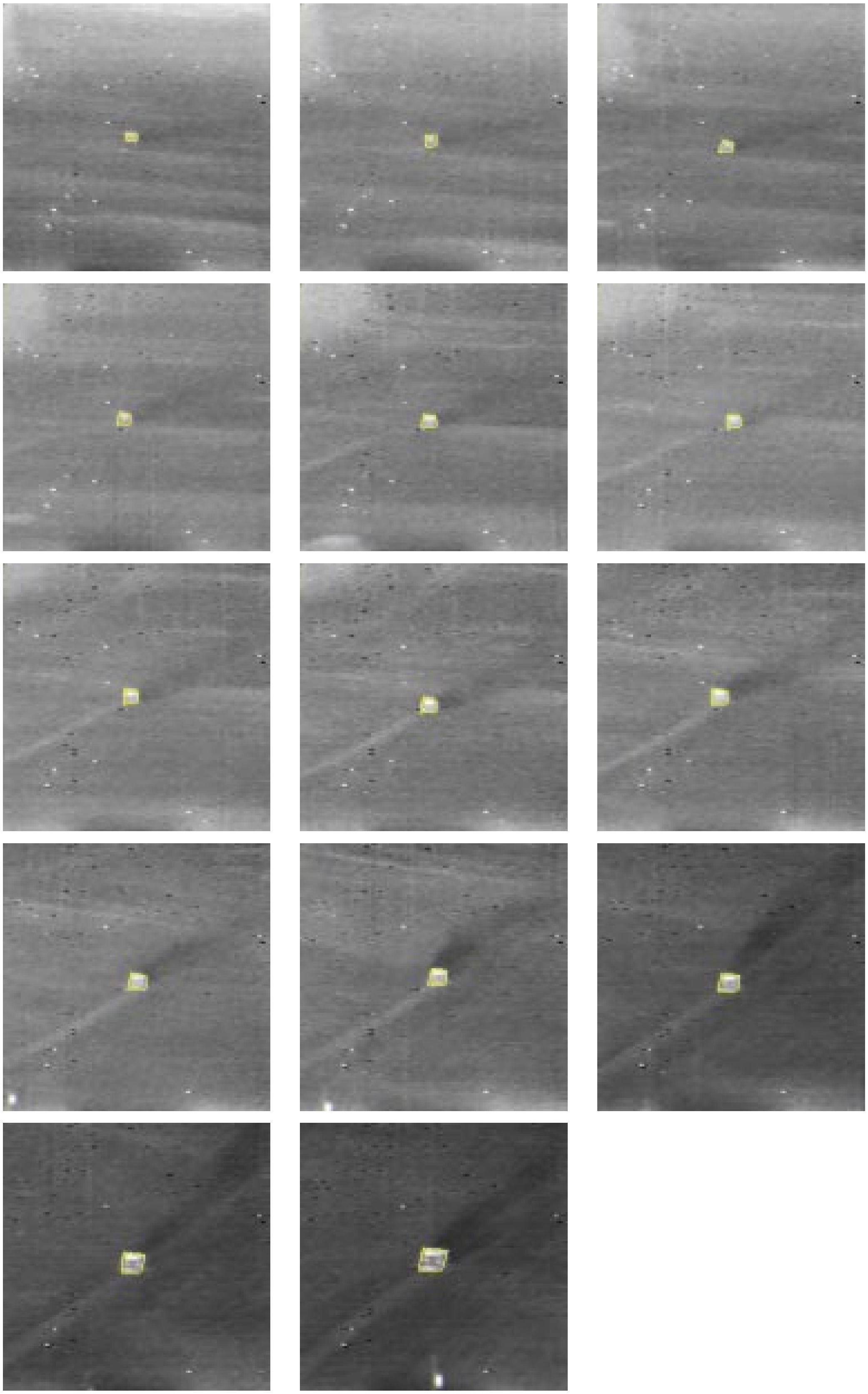


Figure 5.4: Tracking of an IR image sequence II.

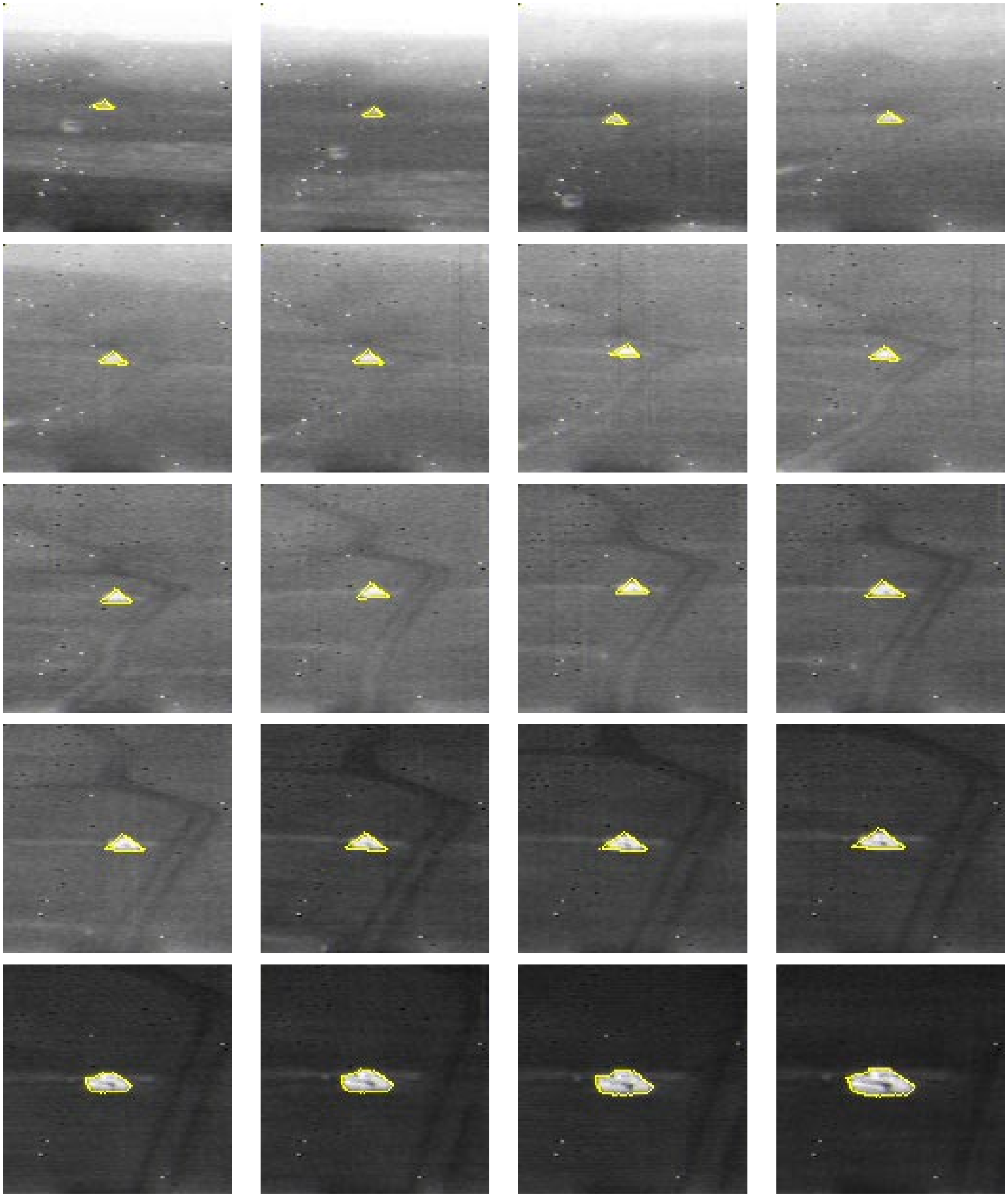


Figure 5.5: Tracking of an IR image sequence III.

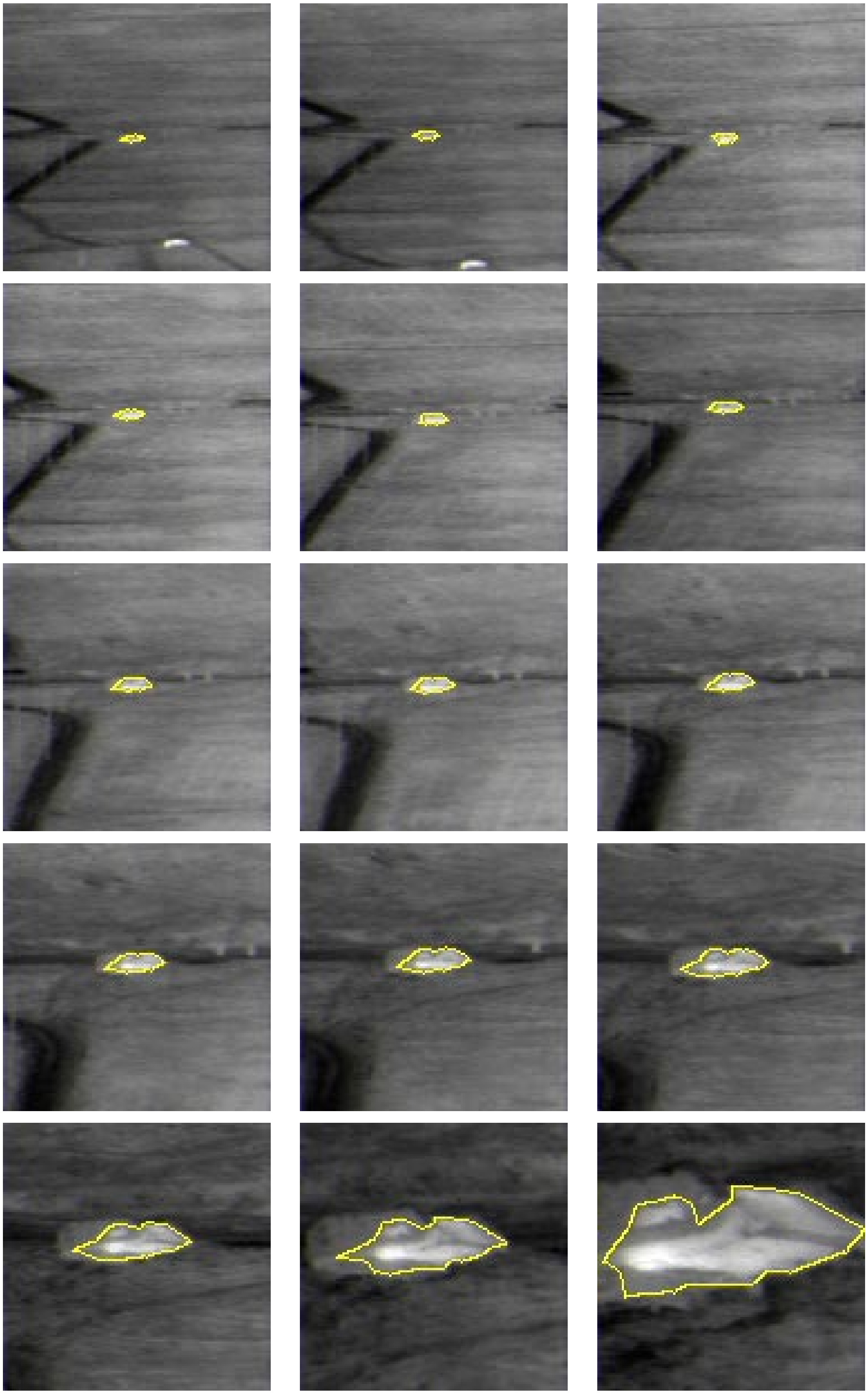


Figure 5.6: Tracking of an IR image sequence IV.

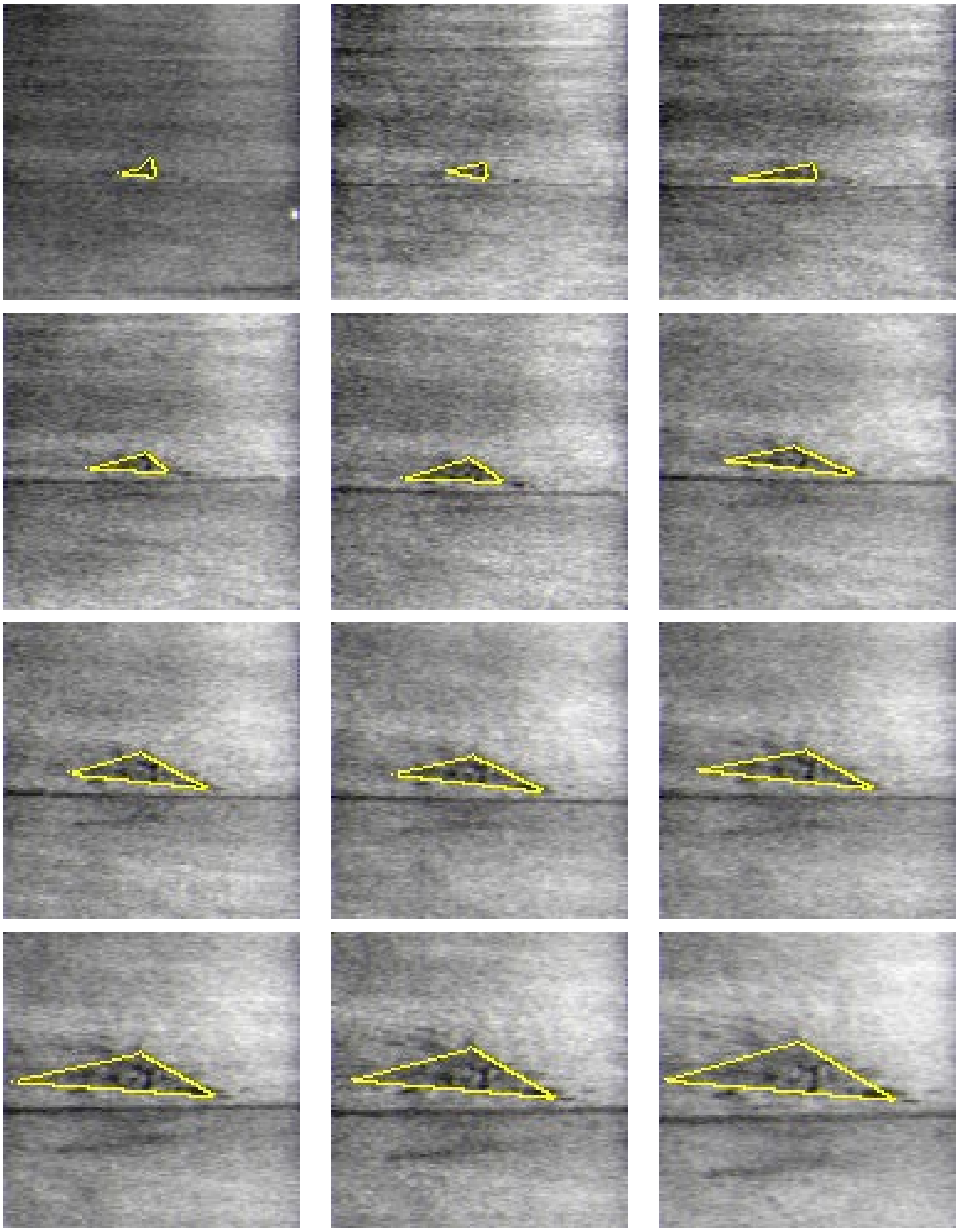


Figure 5.7: Tracking of another IR image sequence V.

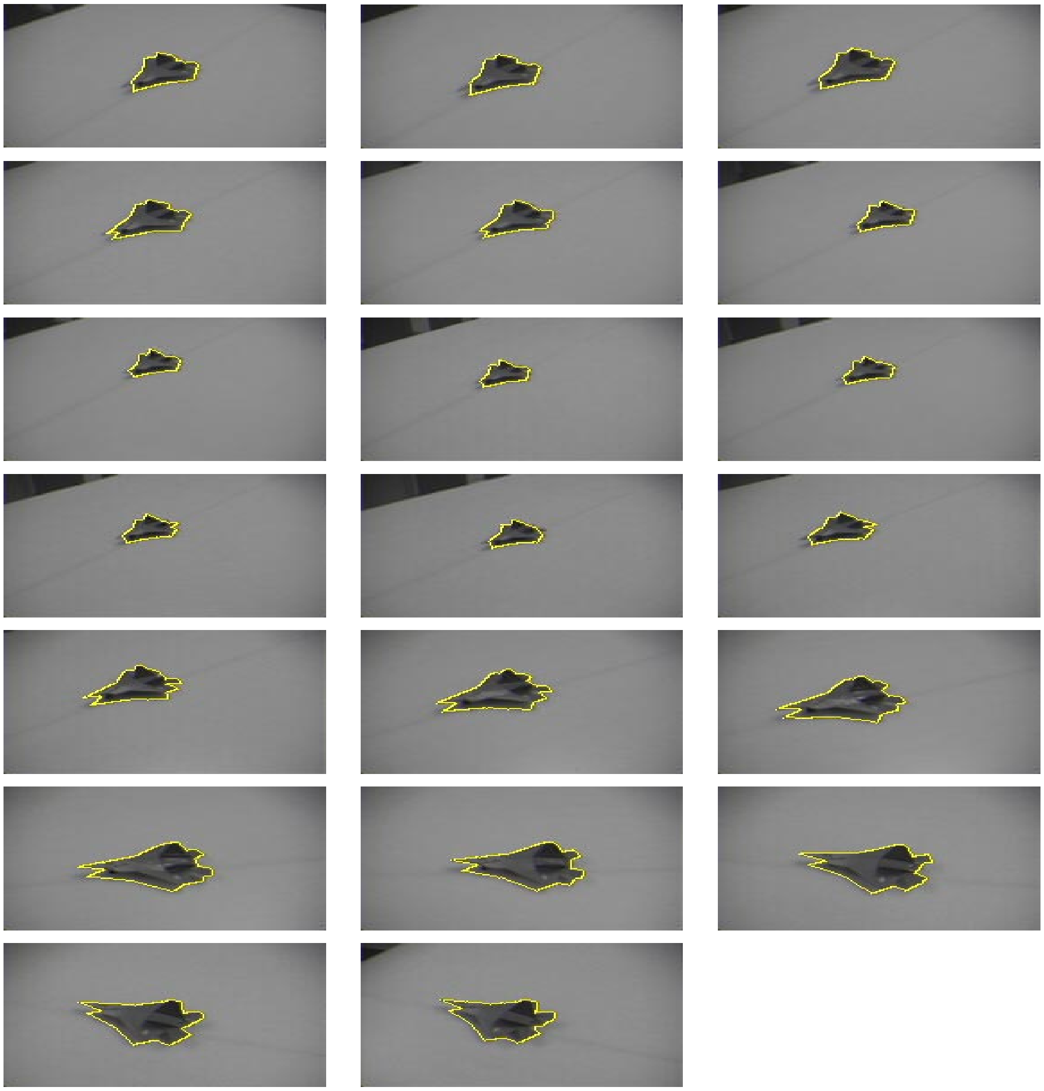


Figure 5.8: Tracking of a toy airplane moving on a conveyor belt (left-right, top-bottom).

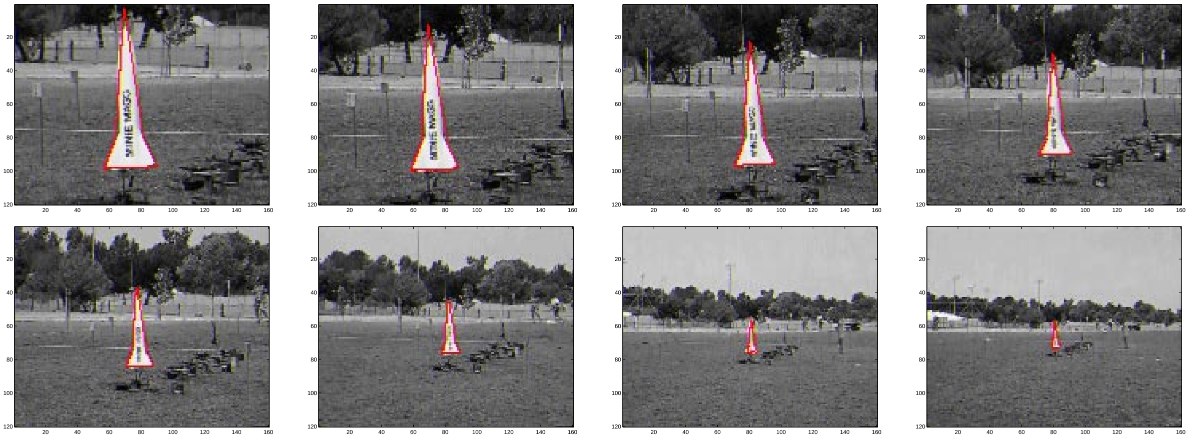


Figure 5.9: Tracking of a model rocket object.

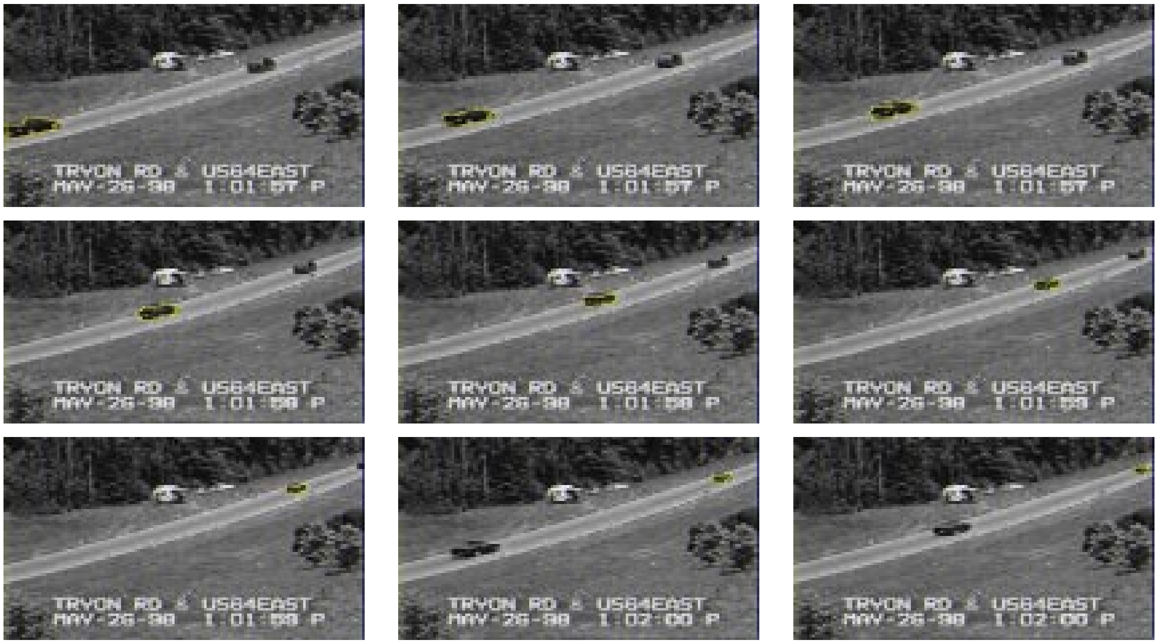


Figure 5.10: Tracking of the black automobile, entering the scene in the top left figure, and leaving the scene in the bottom right figure (sequence provided by courtesy of Alper Unal, Civil Engineering Dept, NCSU).

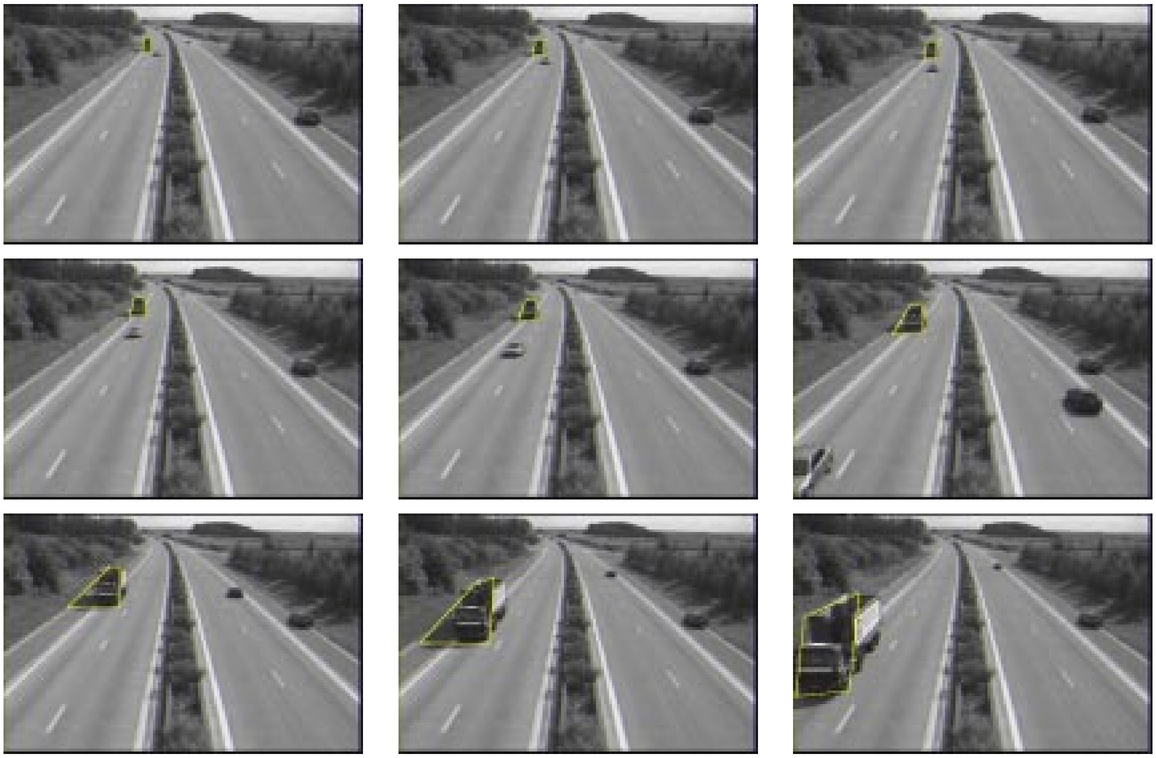


Figure 5.11: Tracking of a truck approaching from the far left (sequence of COST211 project).

5.2 Object Recognition for Visual Information Retrieval

Visual information retrieval refers to the problem of retrieving images or image sequences from a database, which are relevant to a query. Visual information retrieval problem arose as a new topic of research on account of emerging multimedia applications, the availability of large image and video archives, and one's ability of sharing and distributing image/video data over large bandwidth computer networks [15]. First generation visual information retrieval systems allowed querying by strings or text with which capturing, for instance, perceptual elements of a shape or visual features such as the outline of an object is difficult. In the new generation visual information retrieval systems, a more vital description of visual content in agreement with human perception is adopted, where image processing and computer vision techniques are essential for automatic extraction of visual features like color, shape, and texture, from the image data.

The most recent member of the MPEG family, MPEG-7, specifies standardization of visual descriptors (e.g. color, shape, texture, and motion), and description schemes (complex descriptions specifying structure and semantic relationships among descriptors) of multimedia information to allow efficient and fast content-based access and search. Development of visual retrieval systems is hence not only gaining importance but becoming a very important problem as well.

For content-based image retrieval, particularly when objects in an image are of interest, object recognition is a desirable goal, where a decision of whether or not an observed object corresponds to a model in the database has to be made. In the traditional matching approaches, a comparison between the query image and the database images is made, and an ordering according to the measured similarity is output. Selecting a similarity measure is very important, and depends on the representation of perceptual features.

One of the most meaningful descriptions of an object is through its shape, and a simple and efficient way for representing a shape is through a set of features extracted from the image. Features such as the area, the elongatedness, the circularity, the orientation of the major axis describe global shape characteristics, whereas features like corners, characteristic points on the boundary describe local characteristics. Local features are less susceptible to noise due to a partial description they may provide even in the presence of occlusions. Representing a shape through features that describe the boundary of the object, either locally or globally, is a popular method. Considering set of all pixels on the boundary is, however, not efficient as the number of points is usually too large. Reduction of the dimension of the shape's boundary representation by focusing on perceptually salient points (e.g. curvature extrema, corner points), is desirable. Approximating shapes through a polygonal curve has previously been mentioned in shape retrieval techniques [98]. In [98], however, shapes are already input as a polygon to the retrieval system, and polygonal approximation methodology is not discussed in the case of a continuous input shape. A group of interest points, i.e. vertices, form features which are encoded and collectively form a feature index for the query shape. Our approach is taking rather than an indexing approach, a shape matching approach (will be explained in the next subsection), in an image retrieval problem which is more involved than a shape retrieval problem. Also in [98], a discrimination between object recognition problem and similar shape retrieval problem has been made, and the former is defined to require an automatic recognition of an unknown object (without user intervention), whereas the latter one is assumed to require user assistance. We do not make such distinction between the two problems here, and assume an automatic object recognition scenario in both cases.

Model-based Object Recognition using invariants

Existing object recognition systems, which usually address the problem of selecting the object model that best matches the observed image, are referred to as model-based [3, 21, 84, 104]. In these systems, the model objects, and the observed object are described by a set of feature points, whose extraction is the first task to be completed. This is usually followed by a comparison where a quantitative similarity between two shapes, is carried out by a distance measure between their descriptions. The computed shape descriptions should be robust in the context of invariance property so crucial to reliable object recognition.

Geometric invariants are properties of geometric configurations of features which remain unchanged under a certain class of transformations. Geometric characteristics of objects are usually distorted by a perspective projection of a scene from 3D space (real world) to a 2D image plane. For instance, length is an invariant under rigid transform by rotation and translation, but it is changed under a more general class of transformations like affine, or perspective transformation. Parallelism is preserved under an affine transformation (e.g. a square is transformed to an arbitrary parallelogram). On the other hand, collinearity of points is preserved under a projective transformation (equivalent to perspective viewing) [44, 57]. Important invariants of affine transformation are parallel lines, ratio of lengths of parallel line segments, and ratio of areas. The most fundamental invariant of a projective transformation is the cross ratio of four collinear points (a ratio of ratios of lengths on a line).

In model-based vision, affine geometry is often assumed because of the availability of a larger number of invariant features, and the sufficiency of features necessary to uniquely compute model pose under an affine projection rather than a perspective projection.

The crucial step of invariant-based recognition is in the reduction of the huge amount of information in an image to a small number of features which remains invariant to viewing angle, occlusion, lighting conditions which are in turn quintessential to robustness.

There are two approaches to model-based object recognition using geometric invariants.

- **Search-based matching approach:** Invariant features are computed from the observed image and are compared with pre-computed invariant features of all models in the database. This exhaustive search technique is good for small databases, but problematic for large databases.

- **Indexing approach:** The derived transformation-invariant description is used as a key in indexing a database of object models for recognition. In the so-called Geometric Hashing technique [85], objects are represented as sets of geometric features such as points or lines, and their geometric relations are encoded using minimal feature sets to index a hash table prepared beforehand in a learning phase. In the recognition phase, a vote of the hits to entries of the hash table by the feature indices of the observed image are tallied, and a database model object that receives the maximum number of hits is chosen as the corresponding object model for the queried unknown object. These methods are preferred for efficiently accessing very large model databases.

Another classification in image retrieval problem can be made in terms of the homogeneity of the database. The *inter-class* retrieval problem involves distinguishing different classes of objects like airplanes, animals, and cars. The *intra-class* retrieval problem on the other hand involves distinguishing objects from the same class, for instance airplane types F16 and F117. An intra-class indexing application to digital fish databases is given in [4]. Inter-class retrieval may be handled by comparing more global features of a shape, while intra-class retrieval, however, requires more sophisticated approaches to feature comparison.

In the next subsection, we present an intra-class image retrieval study we have carried out as an application of our active polygon model.

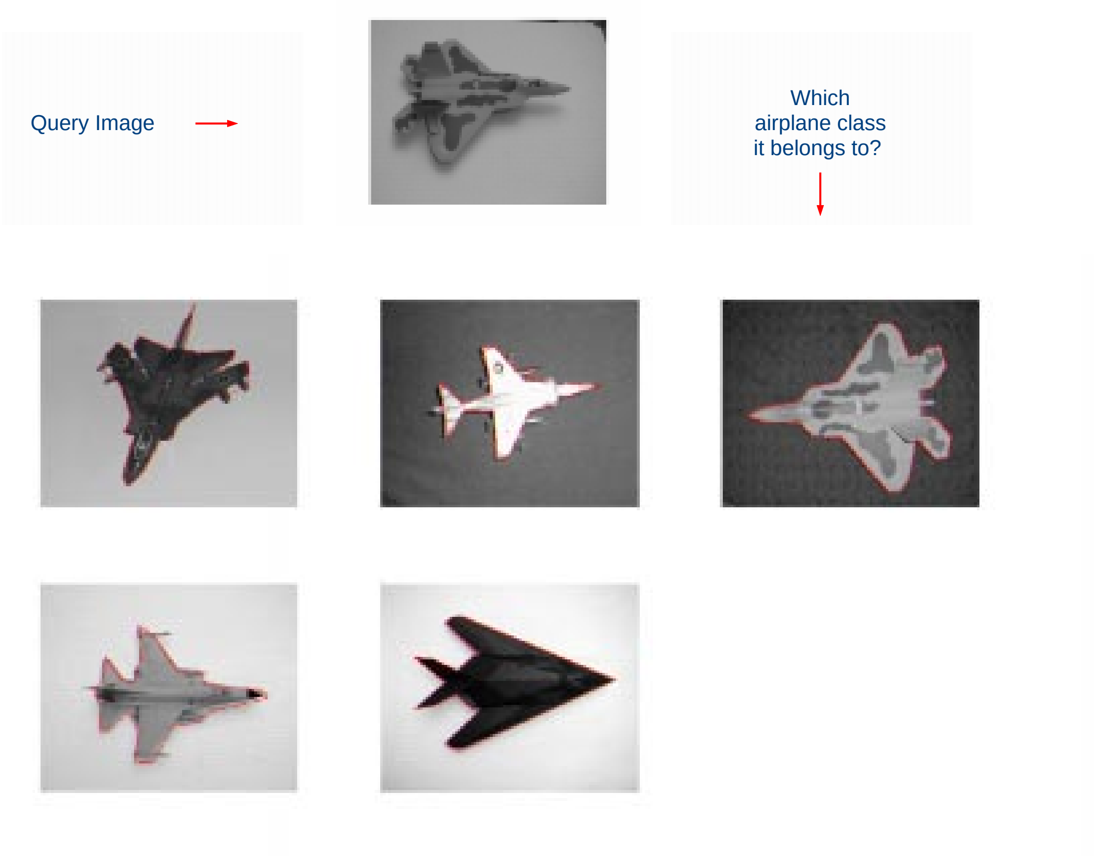


Figure 5.12: Intra-class Image Retrieval Scenario.

5.2.1 Active Polygons for Intra-Class Image Retrieval

Complexity of an intra-class image retrieval problem arises from the fact that objects in a database, e.g. different types of airplanes, and the query image are acquired from different viewing angles, different lighting and background conditions. We performed a recognition experiment on real images as follows. The object database contains pictures of various airplane models. We took photographs of the five different airplane models from a top view (shown in Fig. 5.12), and from different viewing perspectives, and under different lighting conditions, given in Fig. 5.13. This scenario is depicted in Fig. 5.12, where a query airplane image is input to the system, with a question of identifying its membership model. The test image (i.e. the query), is viewed under different lighting and viewing conditions. The key idea is to extract signature features from the test image, and search from the database for the object model that best matches the query in terms of this signature.

Our basis for solving this problem is the active polygon framework we have proposed in Chapter 4. One of the important pre-processing steps for object recognition systems consists of automatically

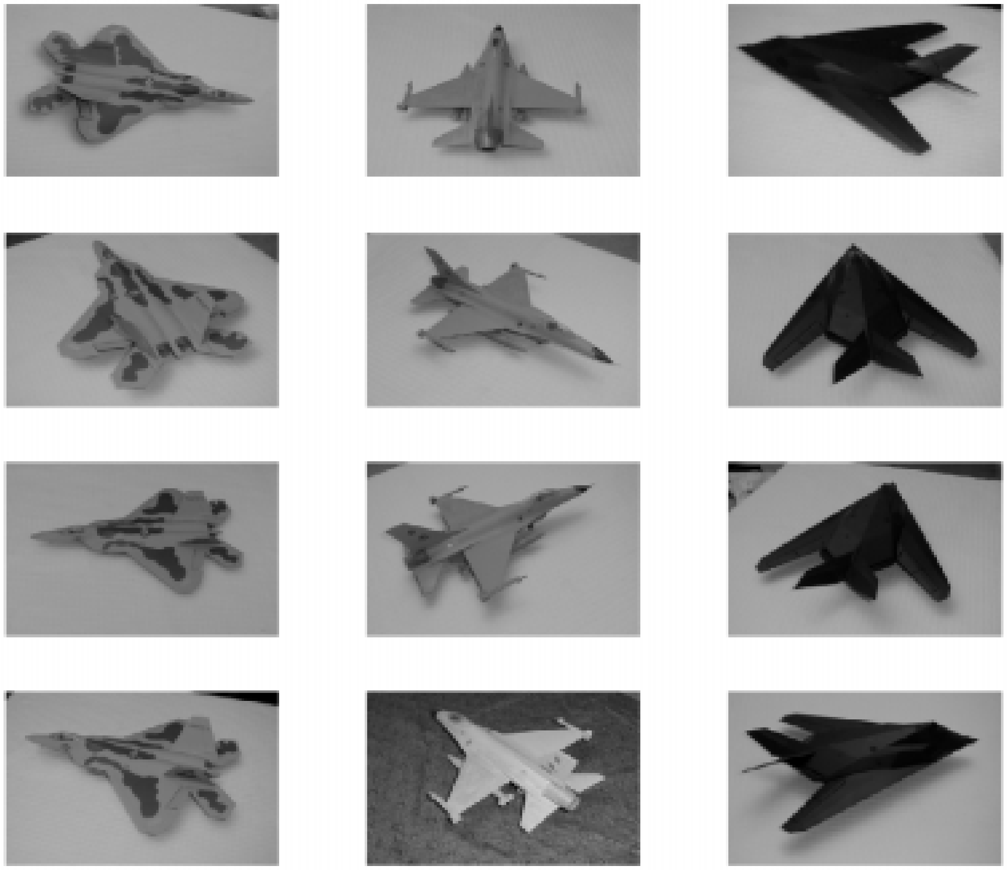


Figure 5.13: Set of airplane images under different conditions.

and systematically extracting features, or the interest points, from an image. We believe that this issue is addressed unsatisfactorily at best in most of the previous object recognition work where manual or supervised extraction was carried out. The active polygons we have developed, on the other hand, can automatically extract important feature points of man-made objects from images. A polygonal contour, for each airplane shape in an image is for instance extracted by our model.

Affine Invariant Parameterization of the Polygon Vertices

The extracted feature points (i.e. the ordered set of vertex points) are used in forming a signature from each shape, and in order to account for some degree of affine invariance, we choose to reparameterize the vertex points by an affine invariant parameterization. An affine transform is a nonsingular linear transformation followed by a translation: $\tilde{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{B}$, where for a 2D representation of shapes, $\boldsymbol{A}$ is a 2×2 matrix, $\boldsymbol{X}$ and $\boldsymbol{B}$ are 2×1 vectors.

A curve $\boldsymbol{C}(p) = (X(p), Y(p))$, $\boldsymbol{C} : [a,b] \to \mathbb{R}^2$ with parameter $p \in [a,b]$ can be reparameterized to a new parameter s via a metric $g(p)$ so that $s(p) = \int_0^p g(\xi)d\xi$. The metric $g(p) = |\det < \boldsymbol{C}, \boldsymbol{C}_p > |$ first proposed in [7], is a first order affine invariant metric for $\boldsymbol{B} = 0$, i.e. the translation component of the affine transform is zero. Note that the affine transform on this metric can be given by $d\tilde{s} = \det < \tilde{\boldsymbol{C}}, \tilde{\boldsymbol{C}}_p > dp = \det < \boldsymbol{A}\boldsymbol{C}, \boldsymbol{A}\boldsymbol{C}_p > dp = \det\boldsymbol{A} < \boldsymbol{C}, \boldsymbol{C}_p > dp$, and the metric is hence linear under an affine transform if $\boldsymbol{B} = 0$.

We note, using the divergence theorem, that this new parameterization

$$t_a = \frac{1}{2}\int_a^b |XY_p - YX_p|dp$$

is indeed the the area enclosed by the contour $\boldsymbol{C}$,

$$\int_R dxdy = \frac{1}{2}\oint_{\partial R = \boldsymbol{C}} < \begin{pmatrix} X \\ Y \end{pmatrix}, \begin{pmatrix} Y_s \\ -X_s \end{pmatrix} > ds.$$

Since the affine transform linearly changes the area with a constant scale $\det\boldsymbol{A}$, a normalization with respect to the total enclosed area of the contour is possible. An initial translation of the polygon to the affine center of mass, as a result, makes this parameterization completely affine invariant, i.e. invariant to translation, rotation, scale, and shear.

In [7], a Fourier transform of the affine-invariant parameterized contour was taken, and invariants were synthesized from the Fourier coefficients. Inspired by this approach, wavelet transform coefficients of the affine-invariant parameterized contour were used as invariants in [3]. In pursuit of exploiting the parsimonious representation provided by our active polygons, we resort to obtain a signature that can be easily matched for a small number of vertices. The cumulative angle function, or turning function [8], measures the angle between the counterclockwise tangent and the x-axis as a function of the arc length s. It increases with left-hand turns and decreases with right-hand turns. For the polygonal contours that we obtain for each image, the turning function is piecewise constant, with increases or decreases at the vertex points. Instead of using the standard turning function as a signature, we propose to use, the turning function of the polygon versus the first order affine arc length (i.e.measure the angle of the counterclockwise tangent as a function of the affine arc length). Although the resulting signature is not completely affine invariant, it will be observed that such a

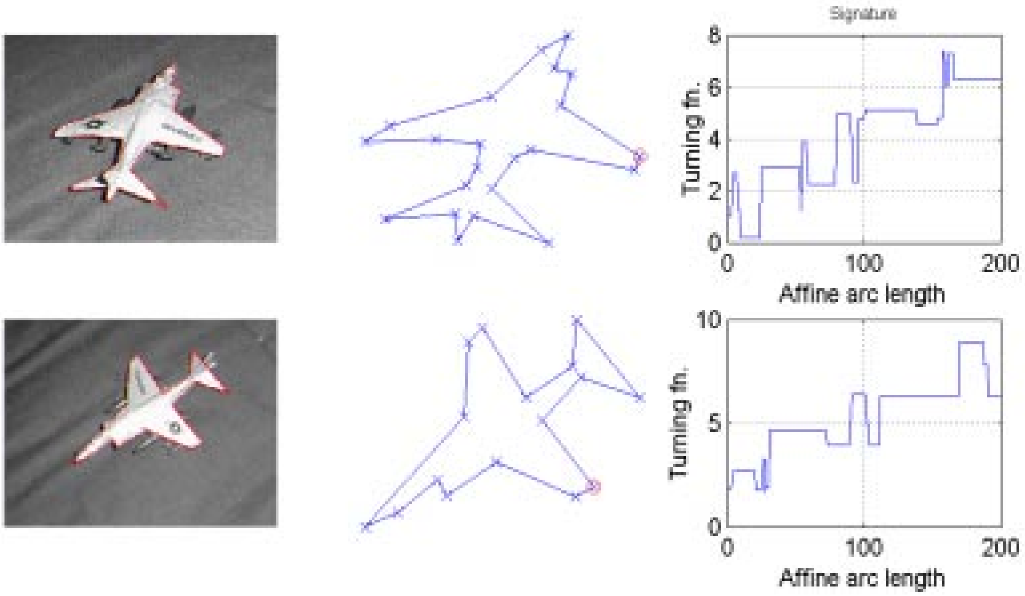


Figure 5.14: An airplane from two different views.

parameterization gives better results in matching polygonal shapes with different poses.

In Figure 5.14, turning function signatures are shown for an airplane from two different views.

In light of the reduced number of vertices of our polygonal representation, one may carry out a cross-correlation on the turning functions of the two polygons in order to account for the starting point variability among shapes. Recall that cross-correlation is a measure of similarity between two different data sets, computed by the sum of the products between the two data sets at different lags. The advantage of our proposed model in using this signature becomes clear as the required number of lags, i.e., the dimension of the cross-correlation function, is very small, (in fact equals the minimum of the number of vertices of the two polygons). For each lag l, the normalized cross-correlation, so-called the cross-correlation coefficient, between two signature functions, Θ_1 and Θ_2, is defined as

$$c_r(l) = \frac{\sum_{t_a}[\Theta_1(t_a) - E\{\Theta_1(t_a)\}][\Theta_2(t_a - l) - E\{\Theta_2(t_a)\}]}{\{\sum_{t_a}[\Theta_1(t_a) - E\{\Theta_1(t_a)\}]^2 \; [\sum_{t_a}\Theta_2(t_a - l) - E\{\Theta_2(t_a)\}]^2\}^{0.5}}. \qquad (5.6)$$

In Figure 5.15, the test image is an F117 whose polygonal representation is extracted by an active polygon, and its signature is computed, and compared to the signatures of the airplanes in the model image set by way of the cross-correlation coefficient. The model object whose signature achieves a maximal cross-correlation with that of the test object's is picked as a top match. The consecutive matches are also ordered with respect to this criterion (Figure 5.16). It can be observed

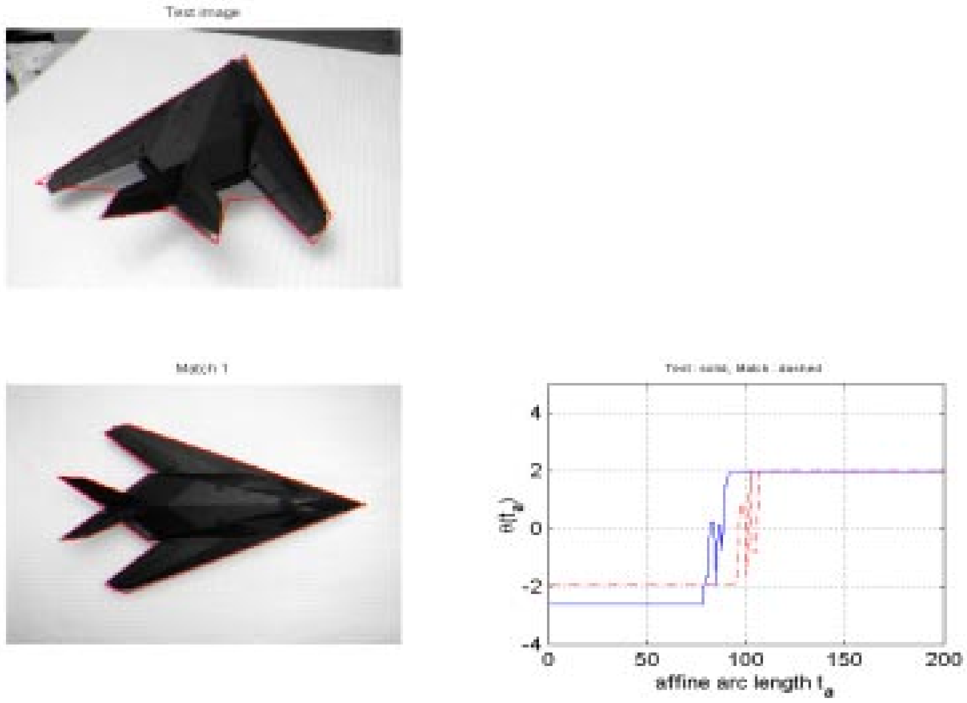


Figure 5.15: A query (an F117) to the model image set, and the best match are shown.

that the top match is indeed the correct model object from the image set.

Another query image, an F16, and the corresponding retrieved object models from the data set is shown in Figure 5.17 and Figure 5.18. The best matching model is an F16.

A Statistical Approach: We also carry out a statistical study when generating the signatures before matching. Due to an inherent variability in real images, the extracted feature set may slightly vary for different initial conditions of an active polygon. To account for this variability, we generated random active polygon initializations on an image, and hence obtained several realizations of the target shape in terms of its feature points (see eight different realizations of an F16 airplane in Figure 5.19). An average signature over the signatures of these realizations may then be obtained, after aligning the shapes by cross-correlating their turning functions vs. standard arc length which is opted to be used in this experiment. The average turning function of the eight realizations in Figure 5.19, computed in this way, is shown in Figure 5.20.

Two retrieval examples using the match over averaged turning functions are shown in Figure 5.21. The correct model from the data set is once again picked in both of the cases.

We note that the application of active polygons to an intra-class image retrieval problem presented above, requires a more thorough performance evaluation, with a larger database and extensive

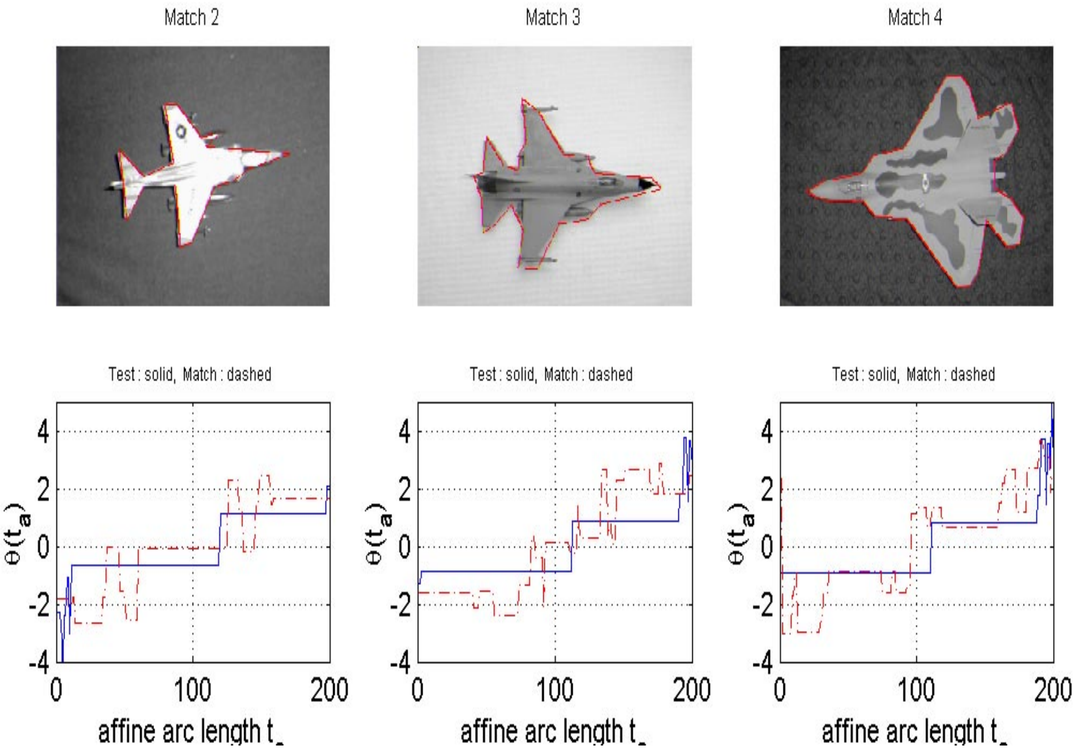


Figure 5.16: The next-to-top matches are shown for the test airplane in Fig. 5.15.

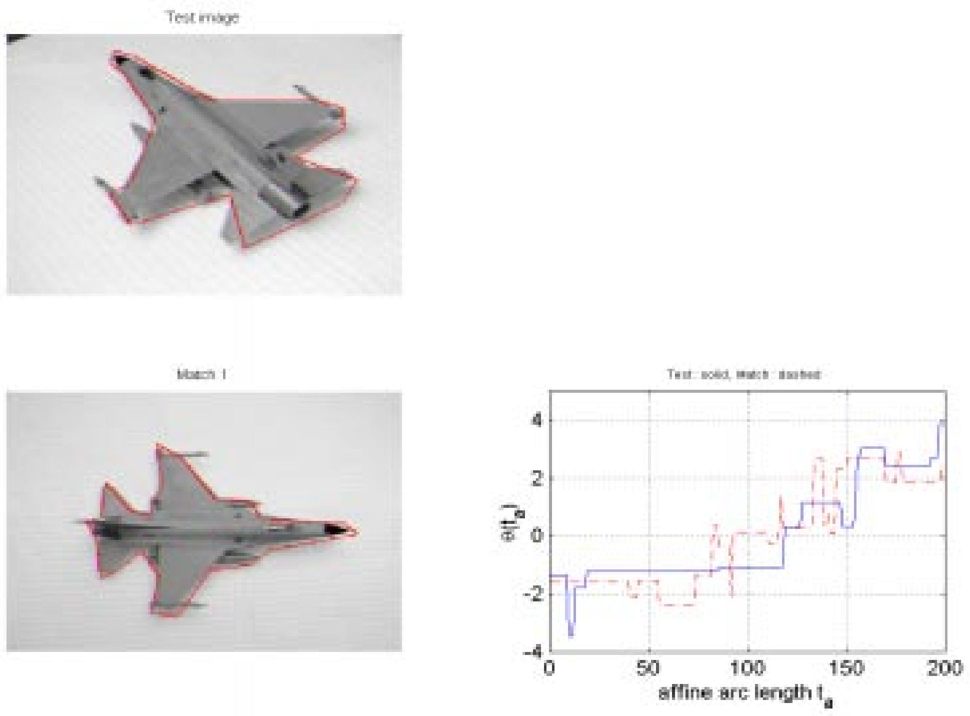


Figure 5.17: A query, which is an F16, to model image set, and a top match are shown.

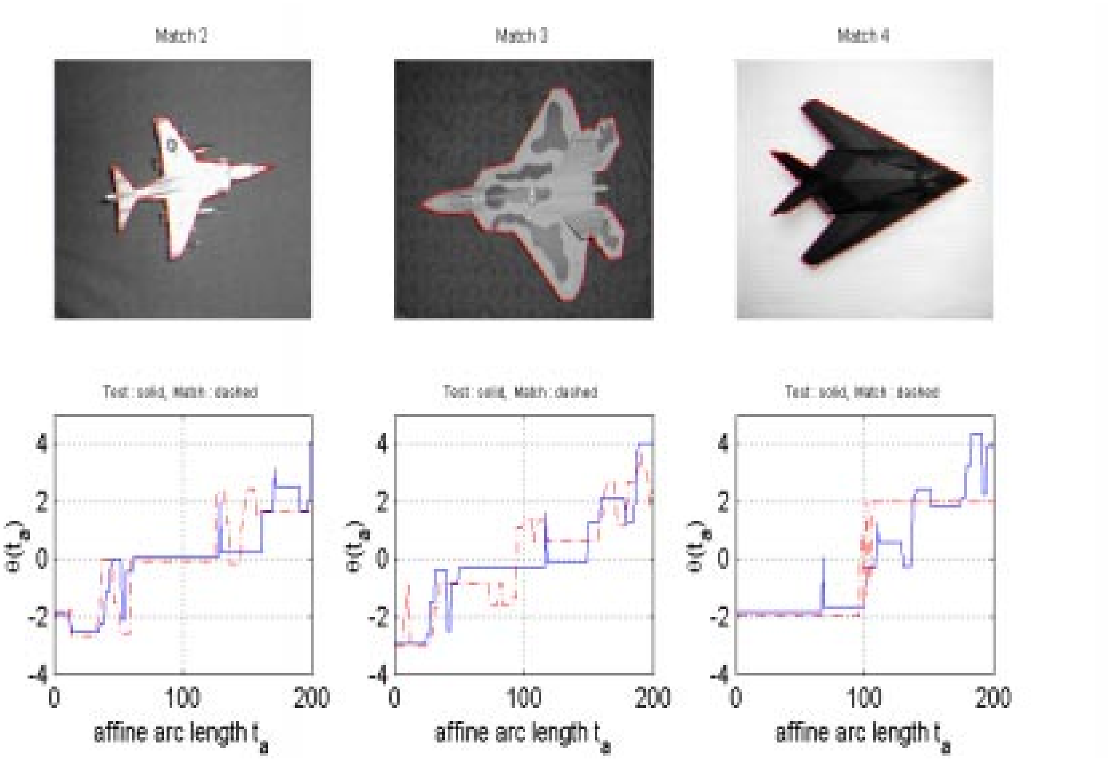


Figure 5.18: The next-to-top matches are shown for the test airplane in Fig. 5.17.

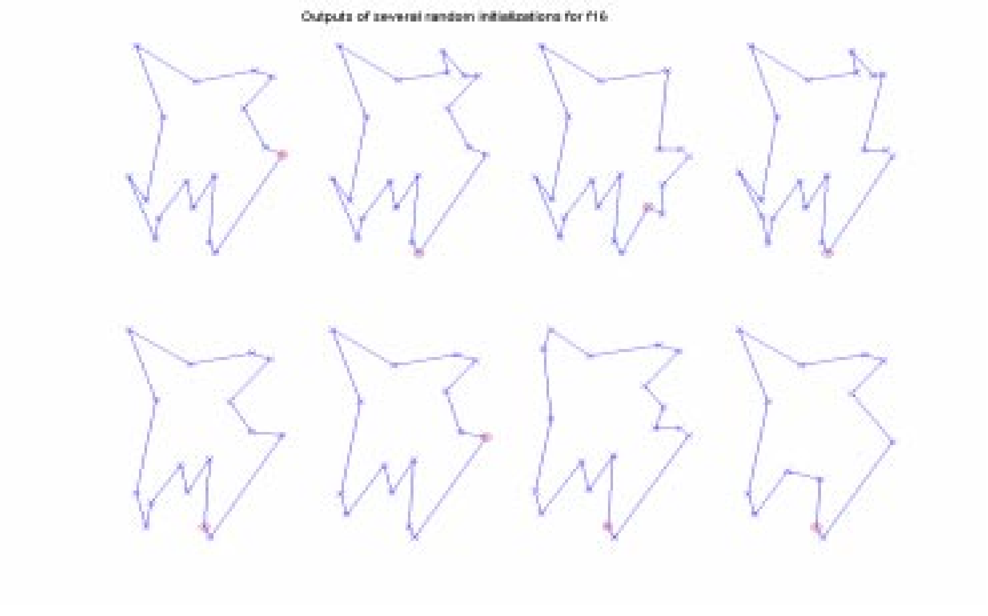


Figure 5.19: Outputs of several random initializations for F16 airplane.

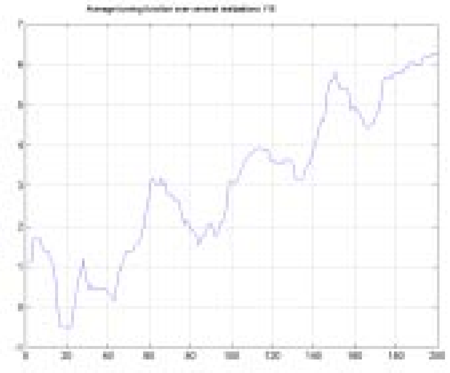
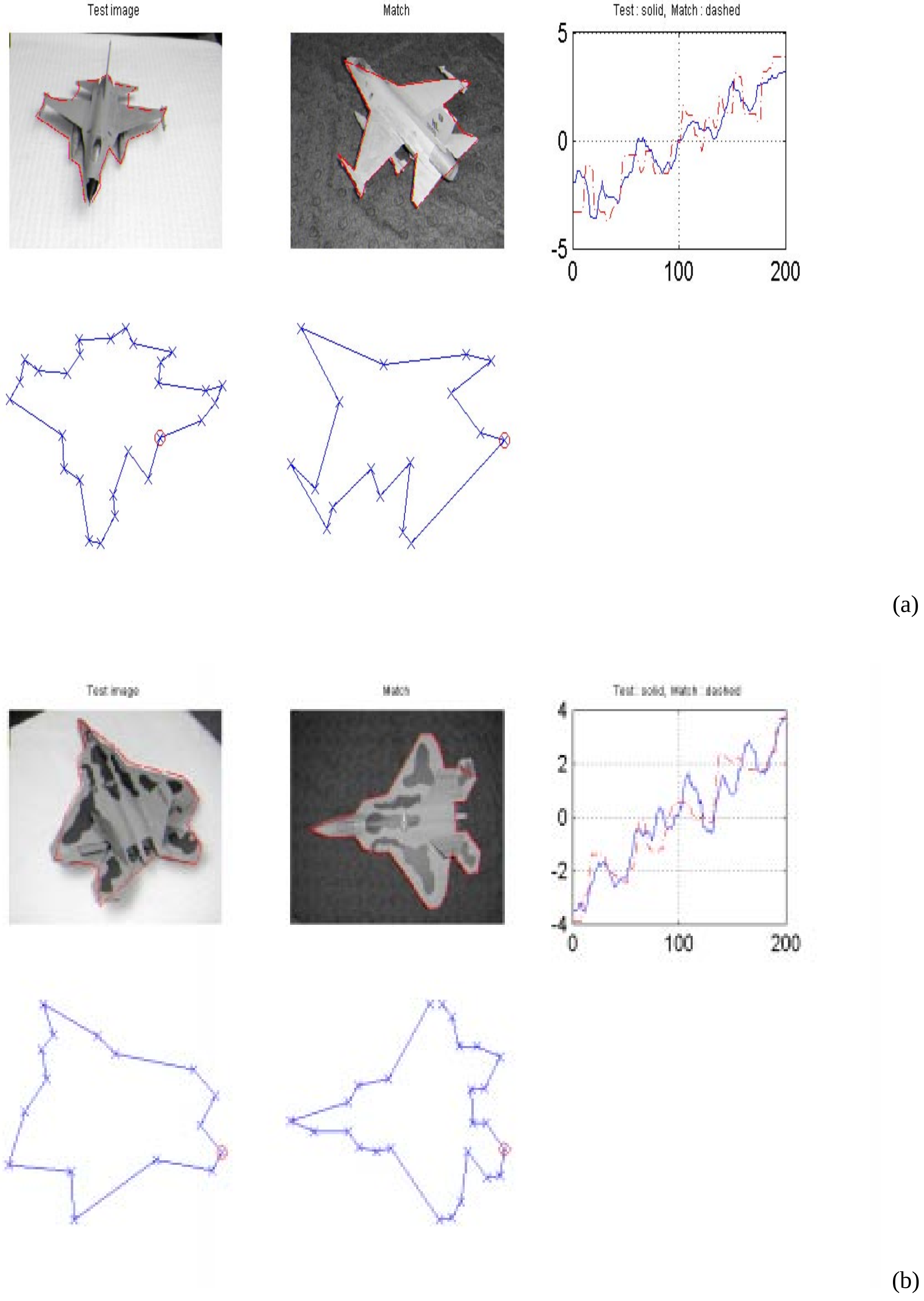


Figure 5.20: Average turning function versus arc length over eight different realizations shown in Fig.5.19.

simulations. This study may be considered as part of a future research direction motivated by this thesis.

5.3 Conclusions

In this chapter, we aimed at demonstrating the flexibility and potential uses of the active polygon framework, developed in Chapter 4, in a varied set of applications in computer vision. First, we presented an object tracking application, which provided a fast way of outlining a moving object through time in a video sequence. An additional gain provided by our technique is the immediate availability of the compact shape description of the tracked object. We also presented an experimental study on applying active polygons to object recognition, which also clearly illustrated the potential of our technique in such applications.



(a)

(b)

Figure 5.21: A query, (a) F16, (b) F22, to model image set, and a top match using averaged turning functions are shown.

Chapter 6

Contributions and Future Research

In this chapter, contributions of this thesis, and the conclusions drawn from the associated research work and the results are presented. Suggestions for future research directions related to this thesis are provided.

6.1 Contributions of the Thesis

In this thesis, we have presented contributions in the development of new image processing algorithms based on curve and polygon evolution models. In Chapter 1, the use of curve evolution techniques for image processing are motivated, and then in Chapter 2, the background material fundamental to the curve evolution concept is given as preliminaries.

6.1.1 Contributions to Curve Evolutions for Nonlinear Filtering

In Chapter 3, we described a new approach to curve and image smoothing through a new class of curve evolution equations designed to preserve prescribed polygonal structures in an image while removing noise. These curve evolutions which are obtained via a directional generalization of the geometric heat equation that circularizes any closed curve, are applied to smooth noisy curves without destroying their significant features on which a prior knowledge is assumed. A contribution in conjunction with these flows is a local stochastic formulation of the geometric heat equation providing an alternative microscopic/macroscopic view of this equation, whose insight led to preset and stabilized polygonal approximations. The theoretical framework of the new class of curve evolutions which leads to a nonlinear filtering along known salient structural directions in an image constitutes the first contribution of this thesis. Additional contributions come from the availability of various applications besides image smoothing. We demonstrated that the designed flows have the ability to morph an input shape into various other shapes, and thus constructed continuous shape transformations have their potential use in shape analysis and shape recognition.

6.1.2 Contributions to Active Contours for Unsupervised Texture Segmentation

Chapter 4 introduced a novel variational framework for an unsupervised texture segmentation problem. The first contribution in this framework, is the new ordinary differential equation model that moves polygon vertices, obtained as the gradient flow of a generic region-based energy functional for an active contour. Under the new evolution model, the evolving contour is a polygon whose every vertex is propagated by an overall speed function integrated along its two adjacent polygon edges. This is a significant improvement over other snakes, geodesic active contours, and region-based active contour techniques. The latter group of techniques which are closer to our model, fuse global information inside and outside the curve with a pointwise local measurement, and is hence not amenable to speed functions that try to capture higher-order statistical features present in most of the textural variations in an image. The second contribution in this set-up is the definition of an information-theoretic measure, based on the Jensen-Shannon divergence, which unfolds the textural information through the underlying probability distributions of the data, moreover using easily computed statistics. Accounting for underlying distributions exactly in order to separate two different textural regions in an image, although desirable, would involve estimation of densities possibly through histograms, would hence be demanding in terms of computations. Utilizing an approximation of Shannon entropy in the definition of the Jensen-Shannon divergence, and in turn in the cost functional, lead to a fast numerical scheme to estimate higher-order characteristics of a probability distribution with minimal effort. In addition, another advantage of adapting Jensen-Shannon divergence as the energy functional of active contours is reflected by its ability of generalization to propagating multiple active contours whose gradient descent equations are derived from a single cost functional, hence leading to a new coupled set of active contour evolution equations. A third contribution in the same active polygon set-up is the new polygon regularizer, which, in contrast to other polygon evolvers, is global and uses a physical basis depending on electrostatics. This is also a significantly new contribution to the field of computer vision in terms of curve evolutions. These three ideas of Chapter 4 together result in a new active contour model, which is significantly different from other snake and active contour methods, and which is successfully applied to more general scenarios of images in terms of texture and to various computer vision applications.

6.1.3 Contributions to Computer Vision

Chapter 5 extended the contribution of Chapter 4 to time-varying image sequences for an object tracking application. The additional contribution here is an efficient and simple tracking methodology which builds on the idea of tracking a relatively small number of vertices by computation of a velocity field at a vertex along its two adjacent edges. We demonstrate through substantiating examples, that the active polygon framework is easily adapted to this scenario. The efficiency of the proposed tracking method comes from the sufficiency of a very sparse set of motion measurements. The motion only along the polygon edges is used, and a coarse estimate at a vertex integrated in a weighted manner the immediately available local measurements, i.e. the component of the velocity orthogonal to an edge. An application to tracking targets in IR image sequences validates the usefulness of the technique. The flexibility of the active polygon framework is also demonstrated in its successful application to an experimental object recognition scenario.

6.2 Future Research Directions

Several interesting research directions motivated by this thesis are discussed next.

6.2.1 Shape Prior on the Curve Evolution Model

In Chapter 3, the flows we designed incorporated a prior knowledge on the salient directions of polygonal shapes whose preservation during smoothing is desired. The form of the functions, $h(\cdot)$ used in the flows were trigonometric, and were based on the unit normal angle, i.e. the local orientation of a point on the curve. They lead to simple n-gone shapes such as triangles, rectangles, hexagons, and so on, on account of the shape prior being used. An open question is how to incorporate other and more general forms of shape priors, for instance global characteristics of a shape such as its size, elongatedness, number of corners and so on. A related question is that through what kind of local and global features, other than orientation, more general shape priors may be integrated by way of the functions utilized in this model. We allude to such a direction in [140, 142], where a variant of the $h(\cdot)$ functions presented in this thesis, was used on the basis of derivative of the local orientation.

6.2.2 A New Application for the Curve Evolution Model

In Chapter 3, we elaborated on the application of the new class of curve evolutions to image and shape smoothing. Another interesting application, hence an interesting future research direction, for these flows when considered as a continuous transformation of one shape into another, is in computer vision and graphics. Morphing in computer graphics transforms the shape of graphical objects such as 2D curves, regions, images, surfaces, and volumes. This may also be considered as a powerful method for shape analysis, where the transformations required to deform one shape into the other provide a way to classify the members of a given shape family.

6.2.3 A New Cost Functional for the Polygon Evolution Model

In Chapter 4, the polygon evolution models we have developed, form a new active contour framework for texture segmentation problem. These models made use of a fast numerical estimation scheme in computation of the Jensen-Shannon divergence, derived from a first-order approximation of a probability density and the Shannon entropy. An avenue of future research would be to define cost functionals which better approximate the underlying probability distributions of regions, or develop more efficient nonparametric estimation techniques of such measures. Adapting the same divergence measure with a Renyi entropy [56] of regions, as a more general measure than the Shannon entropy, would be an interesting investigation into understanding the gains that would be achieved in this case.

6.2.4 Constraint on the Number of Vertices and a Shape Prior on the Polygon Evolution Model

Polygon evolution models developed in Chapter 4, adaptively (by periodically adding and removing vertices) conform to an adequate number of vertices that “minimally” define a target shape in an image. The problem of adding a constraint on the choice of the number of vertices of the polygon to the cost functional of the active contour, would be an interesting future research avenue. The question of putting a penalty on the cost of adding more and more vertices, or a prior distribution on the type of the polygon, its shape, and the number of vertices, and finally coding this prior distribution by a minimum description length criterion would be a research direction that holds promise.

6.2.5 Topological Changes

The active polygon model developed in this thesis makes the assumption of simply connected regions for the image segmentation problem. This assumption is not very restrictive, since in most of the cases, whether in natural images as in zebra, or man-made object images as in airplanes (shown in Chapter 4), texture regions are simply connected. An investigation in how to handle topological changes, i.e. split and merge of a polygon, would be an improvement to capture multiply connected target regions in an image for an extension to more general scenarios.

6.2.6 Further Extensions and Applications for Active Polygons

- Active polygon methodology is extended to an object tracking problem in Chapter 5.1. An improved motion estimation, by using more than the measured orthogonal components of the velocity along two adjacent edges for each vertex, by imposing for instance a smoothness constraint on the velocity field along the contour. This would enhance the object tracking methodology, possibly at a cost of slowing down the performance.

- Designing geometric invariants, fully affine invariant or projective invariant, suitable for the output of the active polygon model would result in a full extension of the object recognition application framework started in Chapter 5.2.

Bibliography

[1] COST211 Project, http://www.iva.cs.tut.fi/COST211.

[2] D. Adalsteinsson and J.A. Sethian. A fast level set method for propagating interfaces. *J. Computational Physics*, 118:269–277, 1995.

[3] R. Alfarez and Y. F. Wang. Geometric and illumination invariants for object recognition. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 21(6):505–536, 1999.

[4] R. Alfarez and Y. F. Wang. Invariant, intra-class retrieval in homogeneous databases. In *IEEE Multimedia Conference*, Japan, 2001.

[5] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel. Axioms and fundamental equations of image processing. *Arch. Rational Mech. Anal.*, 123:199–257, 1993.

[6] L. Alvarez, P.L. Lions, and J.M. Morel. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.*, 29(3):845–866, 1992.

[7] K. Arbter, W. E. Snyder, H. Burkhardt, and G Hirzinger. Application of affine invariant fourier descriptors to recognition of 3-d objects. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 12(7):640–647, 1990.

[8] E.M. Arkin, L.P. Chew, D.P. Huttenlocher, K. Kedem, and J.S.B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 13:209–216, 1986.

[9] J. Badenas, J.M. Sanchiz, and F. Pla. Using temporal integration for tracking regions in traffic monitoring sequences. In *Proc. Int. Conf. on Pattern Recognition*, pages 1125–1128, 2000.

[10] B. Bascle and R. Deriche. Region tracking through image sequences. In *Proc. Int. Conf. on Computer Vision*, pages 302–307, 1995.

[11] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, 1995.

[12] M. O. Berger. How to track efficiently piecewise curved contours with a view to reconstructing 3d objects,. In *Proc. Int. Conf. on Pattern Recognition*, pages 32–36, 1994.

[13] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Stat. Soc. B*, 36:192–236, 1973.

[14] G. Bilbro, W. E. Snyder, S. Garnier, and J. Gault. Mean field annealing: A formalism for constructing gnc-like algorithms. *IEEE Trans. Neural Networks*, 3(1):131–138, 1992.

[15] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann, San Fransisco, CA, 1999.

[16] A. Blake and M. Isard. *Active Contours*. Springer Verlag, London, Great Britain, 1998.

[17] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, Mass., 1987.

[18] H. Blum. Biological shape and visual science. *J. Theor. Biol.*, pages 205–287, 1973.

[19] T.J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 8(1):90–99, 1986.

[20] A. Bruckstein, G. Sapiro, and D. Shaked. Evolutions of planar polygons. *Int. J. Pattern Recognition and Artificial Intell.*, 9(6):991–1014, 1995.

[21] A. Califano and R. Mohan. Multidimensional indexing for recognizing visual shapes. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 16(4):373–392, 1994.

[22] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 8:679–698, 1986.

[23] V. Caselles, F. Catte, T.Coll, and F.Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66:1–31, 1993.

[24] V. Caselles and B. Coll. Snakes in movement. *SIAM Journal on Numerical Analysis*, 33(12):2445–2456, 1996.

[25] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *ICCV95*, pages 694–699, 1995.

[26] A. Chakraborty and J. Duncan. Game-theoretic integration for image segmentation. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 21(1):12–30, 1999.

[27] A. Chakraborty, L. Staib, and J. Duncan. Deformable boundary finding in medical images by integrating gradient and region information. *IEEE Trans. on Medical Imaging*, 15(6):859–870, 1996.

[28] T. Chan and L. Vese. A level set algorithm for minimizing the Mumford-Shah functional in image processing. In *Proc. IEEE Workshop on Variational and Level Set Methods*, 2001.

[29] T.F. Chan and L.A. Vese. An active contour model without edges. In *Int. Conf. Scale-Space Theories in Computer Vision*, pages 141–151, 1999.

[30] S. F. Chang, T. Sikora, and A. Puri (Ed.s). Special issue on MPEG-7. *IEEE Trans. Circuits and Systems on Video Technology*, June 2001.

[31] Y.-G. Chen, Y. Giga, and S. Goto. Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations. *J. Diff. Geom.*, 33:749–786, 1991.

[32] C. Chesnaud, V. Page, and P. Refregier. Improvement in robustness of the statistically independent region snake-based segmentation method of target-shape tracking. *Optics Letters*, 23(7):488–490, 1998.

[33] C. Chesnaud, P. Refregier, and V. Boulet. Statistical region snake-based segmentation adapted to different physical noise models. *PAMI*, 21:1145–1156, 1999.

[34] L. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53:211–218, 1991.

[35] L. Da. F. Costa and L. F. Estrozi. Multiresolution shape representation without border shifting. *Electronics Letters*, 35(21):1829–1830, 1999.

[36] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. New York: Wiley, 1991.

[37] I. J. Cox, J. M. Rehg, and S. Hingorani. A Bayesian multiple hypothesis approach to edge grouping and contour segmentation. *Int. J. Computer Vision*, 11:5–24, 1993.

[38] M.G. Crandall, H. Ishii, and P.L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Math. Soc.*, 27(1):1–67, 1992.

[39] G. Cross and A. K. Jain. Markov random field texture models. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 5:25–39, 1983.

[40] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *Int. J. Computer Vision*, 1(2):167–187, 1987.

[41] M.P. do Carmo. *Differential geometry of curves and surfaces*. Prentice Hall, New Jersey, 1976.

[42] J.G. Dunham. Optimum uniform piecewise linear approximation of planar curves. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 8:67–75, 1986.

[43] H. Farid and E. P. Simoncelli. Optimally rotation-equivariant directional derivative kernels,. In *7th Int'l Conf Computer Analysis of Images and Patterns*, Kiel, Germany, 1997.

[44] O. Faugeras and Q-T. Luong. *The Geometry of Multiple Images*. MIT Press, Cambridge, Mass., 2001.

[45] M.A.T. Figueiredo, J.M.N. Leitao, and A.K. Jain. Unsupervised contour representation and estimation using b-splines and a minimum description length criterion. *IEEE Trans. Image Process.*, 9(6):1075–1087, 2000.

[46] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *Int. J. Computer Vision*, 5(1):77–104, 1990.

[47] S. V. Fogel. The estimation of velocity vector fields from time-varying image sequences. *CVGIP: Image Understanding*, 53(3):253–287, 1991.

[48] H. Freeman. Shape description via the use of critical points. *Pattern Recognition*, 10:159–166, 1978.

[49] Y. Fu, A. T. Erdem, and A. M. Tekalp. Tracking visible boundary of objects using occlusion adaptive motion snake. *IEEE Trans. Image Process.*, 9(12):2051–2060, 2000.

[50] D. Gavrial and L. Davis. 3-d model-based tracking of humans in action: A multi-view approach. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 73–80, 1996.

[51] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 6:721–741, 1984.

[52] J. Gomes. Warping and morphing of graphical objects, Course Notes. *SIGGRAPH '95*, 1995.

[53] J. F. Gomez-Lopera, J. Martinez-Aroza, A. M. Robles-Perez, and R. Roman-Roldan. An analysis of edge detection by using the jensen-shannon divergence. *J. Mathematical Imaging and Vision*, 13:35–56, 2000.

[54] M.A. Grayson. The heat equation shrinks embedded plane curves to round points. *J. Diff. Geom.*, 26:285–314, 1987.

[55] U. Grenander and A. Srivastava. Probability models for clutter in natural images. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 23(4):424–429, 2001.

[56] A. B. Hamza, Y. He, and H. Krim. An information divergence measure for ISAR image registration. In *IEEE Workshop on Statistical Signal Processing*, pages 130–133, 2001.

[57] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[58] Y. He, A. B. Hamza, and H. Krim. Generalized divergence measure for robust image registration. *to appear IEEE Trans. on Signal Processing*, 2002.

[59] D. J. Heeger. Optical flow using spatiotemporal filters. *IJCV*, 1:279–302, 1988.

[60] E. C. Hildreth. Computations underlying the measurement of visual motion. *AI*, 23:309–354, 1984.

[61] H. Hiriyannaiah, G. Bilbro, and W. E. Snyder. Restoration of locally homogeneous images using mean field annealing. *J. Opt. Soc. Am. A*, 6(12):1901–1912, 1989.

[62] B. K. P. Horn. *Robot Vision*. The MIT Press, 1986.

[63] B. K. P. Horn and B. G. Schunck. Determining optical flow. *AI*, 17:185–203, 1981.

[64] A. Hyvarinen. New approximations of differential entropy for independent component analysis and projection pursuit. Technical report, Helsinki University of Technology, 1997.

[65] A. Hyvarinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[66] H. Imai and M. Iri. Computational-geometric methods for polygonal approximations of a curve. *Computer Vision, Graphics, and Image Processing*, 36:31–41, 1986.

[67] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density,. In *Proc. European Conf. Computer Vision*, pages 343–356, 1996.

[68] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[69] E.T. Jaynes. *Information Theory and Statistical Mechanics*, page 181. K.Ford(ed.), New York, 1963.

[70] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Computer Vision*, 1(4):321–331, January 1988.

[71] M. Kendall and A. Stuart. *The Advanced Theory of Statistics*. Charles Griffin & Company, 1958.

[72] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contours. In *Proc. Int. Conf. on Computer Vision*, pages 810–815, 1995.

[73] C. Kim and J. N. Hwang. Fast and automatic video object segmentation and tracking for content-based applications. *IEEE Trans. Circuits and Systems on Video Technology*, 12(2):122–129, 2002.

[74] B.B. Kimia and K. Siddiqi. Geometric heat equation and nonlinear diffusion of shapes and images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1994.

[75] B.B. Kimia, A. Tannenbaum, and S. Zucker. Shapes, shocks, and deformations I. *Int. J. Computer Vision*, (15):189–224, 1995.

[76] B.B. Kimia, A. Tannenbaum, and S.W. Zucker. On the evolution of curves via a function of curvature, I the classical case. *J.Math. Analysis and Applications*, 163:438–458, 1992.

[77] M. W. Koch and R. L. Kashyap. Using polygons to recognize and locate partially occluded objects. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 9(4):483–494, 1987.

[78] G. Koepfler, C. Lopez, and J.M. Morel. A multiscale algorithm for image segmentation by variational methods. *SIAM J. Numer. Anal.*, 31(1):282–299, 1994.

[79] D. Koller, K. Daniilidis, and H. H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *Int. J. Computer Vision*, 10(3):257–281, 1993.

[80] H. Krim and Y. Bao. Smart nonlinear diffusion: A probabilistic approach. Under Review, 2001.

[81] H. Krim and Y. Bao. A stochastic approach to signal denoising. In *ICASSP*, Phoenix, AZ, 1999.

[82] A. Kumar, A. R. Tannenbaum, and G. J. Balas. Optical flow: A curve evolution approach. *IEEE Trans. Image Process.*, 5(4):598–610, 1996.

[83] Y. Kurozumi and W. A. Davis. Polygonal approximation by the minimax method. *Comput. Graph., Image Proc.*, pages 248–264, 1982.

[84] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. Affine invariant model-based object recognition. *IEEE Trans. Robotics and Automation*, 6(5):578–589, 1990.

[85] Y. Lamdan and H. J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proc. Int. Conf. on Computer Vision*, pages 238–249, 1988.

[86] P. D. Lax. Hyperbolic systems of conservation laws and the mathematical theory of shock waves,. In *SIAM Reg. Conf. Series, Lectures in Applied Math*, volume 11, pages 1–47, 1970.

[87] Y. Leclerc. Constructing stable descriptions for image partitioning. *Int. J. Computer Vision*, 3:73–102, 1989.

[88] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhauser, 1992.

[89] F. Leymarie and M. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 15(6):617–634, 1993.

[90] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Trans. Information Theory*, 37(1):145–151, 1991.

[91] D. Lowe. Robust model based motion tracking through the integration of search and estimation. *Int. J. Computer Vision*, 8(2):113–122, 1992.

[92] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*, pages 674–679, 1981.

[93] R. Malladi, J.A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 17(2):158–173, 1995.

[94] B.S. Manjunath and R. Chellappa. Unsupervised texture segmentation using Markov Random Field models. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 13(5):478–482, 1991.

[95] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2d-3d model-based approach,. In *Proc. Int. Conf. on Computer Vision*, pages 262–268, 1999.

[96] D. Marr. *Vision*. W.H. Freeman, San Francisco, 1982.

[97] D. Marr and E. Hildredth. Theory of edge detection. *Proc. R. Soc. London*, B207:187–217, 1980.

[98] R. Mehrotra and J. E. Gary. Similar-shape retrieval in shape data management. *IEEE Computer*, pages 57–62, 1995.

[99] F. G. Meyer and P. Bouthemy. Region-based tracking using affine motion models in long image sequences. *Computer Vision, Graphics, and Image Processing*, 60(2):119–140, 1994.

[100] J.M. Morel and S. Solimini. *Variational Methods in Image Segmentation*. Birkhauser, 1995.

[101] D. Mumford. Elastica and computer vision. In C. Bajaj, editor, *Algebraic Geometry and Its Applications*, pages 491–506. Springer Verlag, New York, 1994.

[102] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *CVPR*, pages 22–26, 1985.

[103] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, San Fransisco, 1985.

[104] J.L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, Cambridge, Mass., 1992.

[105] H. H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 8(5):565–593, 1986.

[106] K.J. O'Connell. Object-adaptive vertex-based shape coding method. *IEEE Trans. Circuits and Systems on Video Technology*, 7:251–255, 1997.

[107] R. L. Ogniewicz and M. Ilg. Voronoi skeletons: Theory and applications. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 63–69, 1992.

[108] B. Oksendal. *Stochastic Differential Equations, An introduction with applications*. Springer, Berlin, 1998.

[109] S. Osher. Riemann solvers, the entropy condition and difference approximations. *SIAM J. Numer. Anal.*, 21:217–235, 1984.

[110] S. Osher and L.I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM J. Numer. Anal.*, 27(4):919–940, 1990.

[111] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on the Hamilton-Jacobi formulation. *J. Computational Physics*, 49:12–49, 1988.

[112] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Syst., Man. Cybernetics*, 9(1):62–66, 1979.

[113] N. Paragios and R. Deriche. Geodesic active regions for motion estimation and tracking. Tech. Report INRIA 1999.

[114] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 22(3):266–280, 2000.

[115] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *Int. J. Computer Vision*, 46(3):223–247, 2002.

[116] D. Peng, S. Osher, B. Merriman, and H-K. Zhao. The geometry of Wulff crystal shapes and its relations with Riemann problems. *Contemporary Mathematics*, 238:251–303, 1999.

[117] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 12(7):629–639, 1990.

[118] S. M. Pizer, A. L. Thall, and D. T. Chen. M-reps: A new object representation for graphics. Technical Report TR99-030, University of North Carolina, Chapel Hill, NC, 17, 1999.

[119] J. Rauch. *Partial Differential Equations*. Springer-Verlag, New York, 1991.

[120] J. Regh and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 612–617, 1995.

[121] R. Ronfard. Region-based strategies for active contour models. In *IJCV*, volume 13, pages 229–251, 1994.

[122] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based removal algorithms. *Physica D*, 60:259–268, 1992.

[123] C. Samson, L. Blanc-Feraud, G. Aubert, and J. Zerubia. A level set method for image classification. In *Int. Conf. Scale-Space Theories in Computer Vision*, pages 306–317, 1999.

[124] G. Sapiro and A. Tannenbaum. Affine invariant scale space. *Int. J. Computer Vision*, 11:25–44, 1993.

[125] G. Sapiro and A. Tannenbaum. On invariant curve evolution and image analysis. *Indiana Univ. Journal of Math.*, 93, 1993.

[126] J.A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*. Cambridge University Press, 1996.

[127] J.A. Sethian. *Level Set Methods and fast marching methods*. Cambridge University Press, 1999.

[128] M.I. Sezan and R.L. Lagendijk (eds.). *Motion Analysis and Image Sequence Processing*. Norwell, MA: Kluwer, 1993.

[129] K. Siddiqi and B.B. Kimia. Parts of visual form: Computational aspects. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1993.

[130] K. Siddiqi, Y.B. Lauziere, A. Tannenbaum, and S.W. Zucker. Area and length minimizing flows for shape segmentation. *IEEE Trans. Image Process.*, 7(3):433–443, 1998.

[131] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multiscale transforms. *IEEE Trans. Information Theory*, 38(2):587–607, 1992.

[132] W. E. Snyder, Y. Han, G. Bilbro, R. Whitaker, and S. Pizer. Image relaxation: Restoration and feature extraction. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 17(6):620–624, 1995.

[133] W. E. Snyder (Ed.). Computer analysis of time varying images, special issue. *IEEE Computer*, 14(8):7–69, 1981.

[134] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS Publishing, 1999.

[135] A. M. Tekalp. *Digital Video Processing*. Prentice Hall, 1995.

[136] B.M. ter Haar Romeny. *Geometry-driven diffusion in computer vision*. Kluwer Academic Publishers, Berlin, 1994.

[137] A. Tsai, A. Yezzi, and A. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Trans. Image Process.*, 10(8):1169–1186, 2001.

[138] F. T. Ulaby. *Fundamentals of Applied Electromagnetics*. Prentice Hall, New Jersey, 1997.

[139] S. Ullman. Analysis of visual motion by biological and computer systems. *IEEE Computer*, 14(8):57–69, 1981.

[140] G. Unal and H. Krim. Segmentation and target recognition in SAR imagery using a level-sets-multiscale-filtering technique. In *Proc. SPIE Aerosense, Wavelet Applications VIII*, volume 4391, Orlando, FL, April 2001.

[141] G. Unal, H. Krim, and A. Yezzi. Stochastic Differential Equations and Geometric Flows. Accepted for publication in *IEEE Trans. on Image Processing*.

[142] G. Unal, H. Krim, and A. Yezzi. Feature-preserving flows: A stochastic differential equation's view. In *Proc. IEEE Int. Conf. on Image Processing*, Vancouver, BC, Canada, September 2000.

[143] G. Unal, H. Krim, and A. Yezzi. Active polygons for object tracking. In *IEEE 1st Int. Conf. 3D Data Processing and Visualization and Transmission*, Padova, Italy, June 2002.

[144] G. Unal, A. Yezzi, and H. Krim. Unsupervised texture segmentation by information-theoretic active polygons. Under review, 2002.

[145] J. Wang and E. Adelson. Representing moving images with layers. *IEEE Trans. Image Process.*, 3(5):625–638, 1994.

[146] L. R. Williams and D. W. Jacobs. Stochastic Completion Fields: A neural model of illusory contour shape and salience. *Neural Computation*, 9:837–858, 1997.

[147] A.P. Witkin. Scale space filtering. In *Proc. Int. Conf. on AI*, pages 1019–1023, Karlsruhe, Germany, 1983.

[148] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum. A geometric snake model for segmentation of medical imagery. *IEEE Trans. on Medical Imaging*, 16(2):199–209, 1997.

[149] A. Yezzi, A. Tsai, and A. Willsky. A statistical approach to snakes for bimodal and trimodal imagery. In *Proc. Int. Conf. on Computer Vision*, pages 898–903, 1999.

[150] A. Yezzi, A. Tsai, and A. Willsky. A fully global approach to image segmentation via coupled curve evolution equations. *Journal of Visual Communication and Image Representation*, 13(1):195–216, 2002.

[151] B.-J. Yun, S.-W. Lee, and S.-D. Kim. Vertex adjustment method using geometric constraint for polygon-based shape coding. *Electronics Letters*, 37:754–755, 2001.

[152] H-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Computational Physics*, 127:179–195, 1996.

[153] S. Zhu and A. Yuille. ”region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation”. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 18:884–900, 1996.

[154] S. C. Zhu, Y. Wu, and D. Mumford. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *Int. J. Computer Vision*, 27(2):107–126, 1998.

APPENDICES

Appendix A

Stochastic Formulation of a Geometric Heat Equation

Let us denote by $u^n(t, \boldsymbol{x}\,)$ the solution to Eq (3.6):

$$\frac{\partial u^n}{\partial t} = \sin^2\left(\tan^{-1}(\frac{u^n_y}{u^n_x})\right)\ u^n_{xx} - \sin\left(2\tan^{-1}(\frac{u^n_y}{u^n_x})\right)\ u^n_{xy} + \cos^2\left(\tan^{-1}(\frac{u^n_y}{u^n_x})\right)\ u^n_{yy},$$

and define

$$\begin{aligned} f_1(u^n_x, u^n_y, u^n_{xx}) &= \sin^2\left(\tan^{-1}(\frac{u^n_y}{u^n_x})\right) u^n_{xx}, \\ f_2(u^n_x, u^n_y, u^n_{xy}) &= \sin\left(2\tan^{-1}(\frac{u^n_y}{u^n_x})\right) u^n_{xy}, \\ f_3(u^n_x, u^n_y, u^n_{yy}) &= \cos^2\left(\tan^{-1}(\frac{u^n_y}{u^n_x})\right) u^n_{yy}. \end{aligned}$$

If we write $u(t, \boldsymbol{x}\,)$ as

$$u(t, \boldsymbol{x}\,) = u^n(t, \boldsymbol{x}\,) + \epsilon u(t, \boldsymbol{x}\,)$$

then $\epsilon u(t, \boldsymbol{x}\,)$ satisfies

$$\begin{aligned} \frac{\partial \epsilon u}{\partial t} &= f_1(u^n_x + \epsilon u_x, u^n_y + \epsilon u_y, u^n_{xx} + \epsilon u_{xx}) - f_1(u^n_x, u^n_y, u^n_{xx}) \\ &- \{f_2(u^n_x + \epsilon u_x, u^n_y + \epsilon u_y, u^n_{xy} + \epsilon u_{xy}) - f_2(u^n_x, u^n_y, u^n_{xy})\} \\ &+ f_3(u^n_x + \epsilon u_x, u^n_y + \epsilon u_y, u^n_{yy} + \epsilon u_{yy}) - f_3(u^n_x, u^n_y, u^n_{yy}). \end{aligned} \tag{A.1}$$

Assuming $f_1(\cdot,\cdot,\cdot), f_2(\cdot,\cdot,\cdot)$ and $f_3(\cdot,\cdot,\cdot)$ are differentiable in their arguments, we can expand $f_1(\cdot,\cdot,\cdot)$ in Taylor series about (u_x^n, u_y^n, u_{xx}^n), $f_2(\cdot,\cdot,\cdot)$ about (u_x^n, u_y^n, u_{xy}^n), and $f_3(\cdot,\cdot,\cdot)$ about (u_x^n, u_y^n, u_{yy}^n). For notational simplicity, let us denote by $\boldsymbol{x_1} = (u_x, u_y, u_{xx})$ and $\boldsymbol{x_1^n} = (u_x^n, u_y^n, u_{xx}^n)$, $\boldsymbol{x_2} = (u_x, u_y, u_{xy})$ and $\boldsymbol{x_2^n} = (u_x^n, u_y^n, u_{xy}^n)$, and $\boldsymbol{x_3} = (u_x, u_y, u_{yy})$ and $\boldsymbol{x_3^n} = (u_x^n, u_y^n, u_{yy}^n)$. If we assume that $\epsilon u(t,\boldsymbol{x})$ is small enough, we can neglect higher order terms and write a linear approximation as

$$\frac{\partial \epsilon u}{\partial t} \approx \epsilon \boldsymbol{x_1} \cdot \nabla f_1(\boldsymbol{x_1^n}) - \epsilon \boldsymbol{x_2} \cdot \nabla f_2(\boldsymbol{x_2^n}) + \epsilon \boldsymbol{x_3} \cdot \nabla f_3(\boldsymbol{x_3^n}) \tag{A.2}$$

$$\begin{aligned}
\frac{\partial \epsilon u}{\partial t} \approx\ & \epsilon (u_x \ \ u_y \ \ u_{xx}) \cdot \left(\begin{array}{l} 2\sin\left(\tan^{-1}(\frac{u_y^n}{u_x^n})\right)\cos\left(\tan^{-1}(\frac{u_y^n}{u_x^n})\right) \frac{1}{1+(\frac{u_y^n}{u_x^n})^2} \frac{-u_y^n}{(u_x^n)^2}\, u_{xx}^n \\ 2\sin\left(\tan^{-1}(\frac{u_y^n}{u_x^n})\right)\cos\left(\tan^{-1}(\frac{u_y^n}{u_x^n})\right) \frac{1}{1+(\frac{u_y^n}{u_x^n})^2} \frac{1}{u_x^n}\, u_{xx}^n \\ \sin^2\left(\tan^{-1}(\frac{u_y^n}{u_x^n})\right) \end{array} \right) \\
- \ & \epsilon (u_x \ \ u_y \ \ u_{xy}) \cdot \left(\begin{array}{l} 2\cos\left(2\tan^{-1}(\frac{u_y^n}{u_x^n})\right) \frac{1}{1+(\frac{u_y^n}{u_x^n})^2} \frac{-u_y^n}{(u_x^n)^2}\, u_{xy}^n \\ 2\cos\left(2\tan^{-1}(\frac{u_y^n}{u_x^n})\right) \frac{1}{1+(\frac{u_y^n}{u_x^n})^2} \frac{1}{u_x^n}\, u_{xy}^n \\ \sin\left(2\tan^{-1}(\frac{u_y^n}{u_x^n})\right) \end{array} \right) \\
+ \ & \epsilon (u_x \ \ u_y \ \ u_{yy}) \cdot \left(\begin{array}{l} 2\cos\left(\tan^{-1}(\frac{u_y^n}{u_x^n})\right)\left(-\sin\left(\tan^{-1}(\frac{u_y^n}{u_x^n})\right)\right) \frac{1}{1+(\frac{u_y^n}{u_x^n})^2} \frac{-u_y^n}{(u_x^n)^2}\, u_{yy}^n \\ 2\cos\left(\tan^{-1}(\frac{u_y^n}{u_x^n})\right)\left(-\sin\left(\tan^{-1}(\frac{u_y^n}{u_x^n})\right)\right) \frac{1}{1+(\frac{u_y^n}{u_x^n})^2} \frac{1}{u_x^n}\, u_{yy}^n \\ \cos^2\left(\tan^{-1}(\frac{u_y^n}{u_x^n})\right) \end{array} \right)
\end{aligned} \tag{A.3}$$

Defining the corresponding nominal angle $\theta^n(t,\boldsymbol{x}) = \tan^{-1}(\frac{u_y^n(t,\boldsymbol{x})}{u_x^n(t,\boldsymbol{x})})$, and re-arranging the terms of Eq. (A.3), we get the linearized version of the geometric heat equation around a nominal value:

$$\begin{aligned}
\frac{\partial u(t,\boldsymbol{x})}{\partial t} \ &\approx\ \mathcal{A}_{\text{GHElin}} u(t,\boldsymbol{x}) \\
&= \sin^2(\theta^n(t,\boldsymbol{x}))\, u_{xx}(t,\boldsymbol{x}) - \sin(2\theta^n(t,\boldsymbol{x}))\, u_{xy}(t,\boldsymbol{x}) + \cos^2(\theta^n(t,\boldsymbol{x}))\, u_{yy}(t,\boldsymbol{x}) \\
&+ \ \ c\ \left(-u_y^n(t,\boldsymbol{x})\, u_x(t,\boldsymbol{x}) + u_x^n(t,\boldsymbol{x})\, u_y(t,\boldsymbol{x})\right),
\end{aligned} \tag{A.4}$$

where $c = \frac{1}{(u_x^n)^2 + (u_y^n)^2}\left[\sin(2\theta^n)(u_{xx}^n - u_{yy}^n) - \cos(2\theta^n) 2 u_{xy}^n\right]$.

Appendix B

Derivation of ODEs for vertex motion

Let us define $\boldsymbol{C}_v$ for vertex $\boldsymbol{V}_i$ by

$$\boldsymbol{C}_v(p, \boldsymbol{V}) = \begin{cases} \big(p-(i-1)\big)\boldsymbol{e} & \text{for} \quad i-1 \leq p \leq i \\ \big(1-(p-i)\big)\boldsymbol{e} & \text{for} \quad i \leq p \leq i+1 \\ 0 & \text{for} \quad |p-i| \geq 1 \end{cases}$$

where $\boldsymbol{e} = (\boldsymbol{e}_x, \boldsymbol{e}_y)^T$ denote the standard bases for $\mathbb{R}^2$. We now write our energy as a function of the vertices V : $E(V) = \int_0^n \langle F, N\rangle \, \|C_p\| dp = \int_0^n \langle F, JC_p\rangle \, dp$, and compute its partial derivative with respect to one of the vertex coordinates v where either $v = x_i$ or $v = y_i$ for some $1 \leq i \leq n$, as $E_v(V) = \int_0^n f\langle C_v, JC_p\rangle \, dp$. Substituting the particular forms of $\boldsymbol{C}$, $\boldsymbol{C}_p$ (Eq.(4.4)), and $\boldsymbol{C}_v$ (the latter denoting either $\boldsymbol{C}_{x_i}$ or $\boldsymbol{C}_{y_i}$ for some vertex i) into this expression yields

$$\begin{aligned} E_v = & \int_0^n f\big(L(p-\lfloor p\rfloor, \boldsymbol{V}_{\lfloor p\rfloor}, \boldsymbol{V}_{\lfloor p\rfloor+1})\big) \, \big\langle \boldsymbol{C}_v, J(\boldsymbol{V}_{\lfloor p\rfloor+1} - \boldsymbol{V}_{\lfloor p\rfloor})\big\rangle \, dp \\ = & \sum_{k=0}^{n-1} \int_0^1 f\big(L(p, \boldsymbol{V}_k, \boldsymbol{V}_{k+1})\big) \, \big\langle \boldsymbol{C}_v(p+k), J(\boldsymbol{V}_{k+1} - \boldsymbol{V}_k)\big\rangle \, dp \\ = & \int_0^1 f\big(L(p, \boldsymbol{V}_{i-1}, \boldsymbol{V}_i)\big)\big\langle \boldsymbol{C}_v(p+i-1), J(\boldsymbol{V}_i - \boldsymbol{V}_{i-1})\big\rangle \, dp \\ & + \int_0^1 f\big(L(p, \boldsymbol{V}_i, \boldsymbol{V}_{i+1})\big)\big\langle \boldsymbol{C}_v(p+i), J(\boldsymbol{V}_{i+1} - \boldsymbol{V}_i)\big\rangle dp \\ = & \big\langle \boldsymbol{e}, J(\boldsymbol{V}_i - \boldsymbol{V}_{i-1})\big\rangle \int_0^1 pf\big(L(p, \boldsymbol{V}_{i-1}, \boldsymbol{V}_i)\big) \, dp \\ & + \big\langle \boldsymbol{e}, J(\boldsymbol{V}_{i+1} - \boldsymbol{V}_i)\big\rangle \int_0^1 (1-p) f\big(L(p, \boldsymbol{V}_i, \boldsymbol{V}_{i+1})\big) dp \end{aligned}$$

where $\boldsymbol{e} = \boldsymbol{e}_x$ if $v = x_i$ or $\boldsymbol{e} = \boldsymbol{e}_y$ if $v = y_i$. If we introduce a time variable t and evolve both coordinates x_i and y_i in the gradient directions given above, and denoting the corresponding $J(\boldsymbol{V}_i - \boldsymbol{V}_{i-1}) = \boldsymbol{N}_{i,i-1}$, we obtain the following gradient flow for the vertex $\boldsymbol{V}_i$

$$\frac{\partial \boldsymbol{V}_i}{\partial t} = \int_0^1 p f\big(L(p, \boldsymbol{V}_{i-1}, \boldsymbol{V}_i)\big)\, dp\, \boldsymbol{N}_{i,i-1} + \int_0^1 (1-p) f\big(L(p, \boldsymbol{V}_i, \boldsymbol{V}_{i+1})\big)\, dp\, \boldsymbol{N}_{i+1,i}.$$

Appendix C

Derivation of the Gradient Flow for Varying Priors proportional to areas

Rewriting here Eq.(4.19), (note that $A = |R_u| + |R_v|$),

$$\begin{aligned}\frac{\partial \boldsymbol{C}}{\partial t} = \nabla \widehat{JS}_{a,2} &= \sum_{j=1}^{m} \frac{1}{2} \frac{|R_u||R_v|}{A^2} \left(2(u_j - v_j)(\nabla u_j - \nabla v_j)\right) \boldsymbol{N}_u \\ &+ \frac{1}{2} \frac{|R_v|}{A^2} (u_j - v_j)^2 \boldsymbol{N}_u - \frac{1}{2} \frac{|R_u|}{A^2} (u_j - v_j)^2 \boldsymbol{N}_u, \end{aligned} \tag{C.1}$$

it may be rewritten as follows:

$$\begin{aligned}\frac{\partial \boldsymbol{C}}{\partial t} &= \frac{\boldsymbol{N}_u}{2A^2} \sum_{j=1}^{m} (u_j - v_j) \Bigg(\frac{2|R_u||R_v|(G_j(I) - u_j)}{|R_u|} + \frac{2|R_u||R_v|(G_j(I) - v_j)}{|R_v|} \\ &\quad + (|R_v| - |R_u|)(u_j - v_j) \Bigg) \\ &= \frac{\boldsymbol{N}_u}{2A^2} \sum_{j=1}^{m} (u_j - v_j) \left(2|R_v|(G_j(I) - u_j) + 2|R_u|(G_j(I) - v_j) + (|R_v| - |R_u|)(u_j - v_j)\right) \\ &= \frac{\boldsymbol{N}_u}{2A^2} \sum_{j=1}^{m} (u_j - v_j) \left(2|R_v|G_j(I) - 2|R_v|u_j + 2|R_u|G_j(I) - 2|R_u|v_j + (|R_v| - |R_u|)(u_j - v_j)\right) \\ &= \frac{\boldsymbol{N}_u}{2A^2} \sum_{j=1}^{m} (u_j - v_j) \left(2(|R_u| + |R_v|)G_j(I) - |R_v|u_j - |R_u|v_j - |R_u|u_j - |R_v|v_j\right) \\ &= \frac{\boldsymbol{N}_u}{2A} \sum_{j=1}^{m} (u_j - v_j)(G_j(I) - u_j + G_j - v_j) \\ &= \frac{\boldsymbol{N}_u}{2A} \sum_{j=1}^{m} [-(G_j(I) - u_j)^2 + (G_j(I) - v_j)^2]. \end{aligned} \tag{C.2}$$

It can be observed that Eq.(C.2) is the gradient descent flow for the following energy functional

$$E = \frac{1}{2A} \sum_{j=1}^{m} \int_{R_u} (G_j(I) - u_j)^2 dx dy + \int_{R_v} (G_j(I) - v_j)^2 dx dy, \qquad \text{(C.3)}$$

which is a generalized form of the energy functional proposed by Chan and Vese [29].

Appendix D

Derivation of the Electric Field at a point exerted by a line charge

The integral we want to evaluate is given by

$$\begin{aligned} \boldsymbol{E}_{ab}(\boldsymbol{x}') &= \int_{\boldsymbol{a}}^{\boldsymbol{b}} \frac{\boldsymbol{x}' - \boldsymbol{x}}{||\boldsymbol{x}' - \boldsymbol{x}||^3} \lambda dx \qquad \text{(D.1)} \\ &= k_c L \int_0^1 \frac{(\boldsymbol{x}' - \boldsymbol{a}) + t(\boldsymbol{a} - \boldsymbol{b})}{||(\boldsymbol{x}' - \boldsymbol{a}) + t(\boldsymbol{a} - \boldsymbol{b})||^3} dt. \qquad \text{(D.2)} \end{aligned}$$

By making change of variables, $\boldsymbol{u} = \boldsymbol{x}' - \boldsymbol{a}$, and $\boldsymbol{v} = \boldsymbol{a} - \boldsymbol{b}$, the integral above can be written as

$$\boldsymbol{E}_{ab} = \int_0^1 \frac{\boldsymbol{u} + t\boldsymbol{v}}{||\boldsymbol{u} + t\boldsymbol{v}||^3} dt = \int_0^1 \boldsymbol{f}'(t) dt, \qquad \text{(D.3)}$$

where we denoted the integrand by a vector $\boldsymbol{f}'(t)$, a function of t, which is the derivative of the solution $\boldsymbol{f}(t)$ w.r.t t. We expect the solution $\boldsymbol{f}(t)$, to this integral equation, to be of the form

$$\boldsymbol{f}(t) = \frac{(a\boldsymbol{u} + b\boldsymbol{v}) + t(c\boldsymbol{u} + d\boldsymbol{v})}{||\boldsymbol{u} + t\boldsymbol{v}||}, \qquad \text{(D.4)}$$

where a, b, c, and d are scalars to be estimated. Taking the derivative of $\boldsymbol{f}(t)$ w.r.t. t, we obtain (note that in the following, for simplicity we denote the inner product of a vector by itself $\boldsymbol{v}^T\boldsymbol{v}$ by $\boldsymbol{v}^2$, and the inner product by $\cdot$)

$$\boldsymbol{f}'(t) = \frac{(c\boldsymbol{u} + d\boldsymbol{v})(\boldsymbol{v}^2 + 2t\boldsymbol{u}\boldsymbol{v} + t^2\boldsymbol{v}^2) - [(a\boldsymbol{u} + b\boldsymbol{v}) + t(c\boldsymbol{u} + d\boldsymbol{v})](\boldsymbol{u}\boldsymbol{v} + t\boldsymbol{v}^2)}{||\boldsymbol{u} + t\boldsymbol{v}||^3}. \qquad \text{(D.5)}$$

The numerator above can hence be equated to the numerator in the original integral, i.e. $\boldsymbol{u} + t\boldsymbol{v}$. This leads to the following two equalities:

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{u}\,(c\boldsymbol{u}^2 - a(\boldsymbol{u}\,\boldsymbol{v})) + \boldsymbol{v}\,(d\boldsymbol{u}^2 - b(\boldsymbol{u}\,\boldsymbol{v})) \\ \boldsymbol{v} &= \boldsymbol{u}\,(c(\boldsymbol{u}\,\boldsymbol{v}) - a\boldsymbol{v}^2) + \boldsymbol{v}\,(d(\boldsymbol{u}\,\boldsymbol{v}) - b\boldsymbol{v}^2) \end{aligned} \tag{D.6}$$

Equating the corresponding coefficients of $\boldsymbol{u}$ and $\boldsymbol{v}$ to 0 and 1 as required leads to the following system of linear equations in four unknowns:

$$\begin{bmatrix} -\boldsymbol{u}\,\boldsymbol{v} & 0 & \boldsymbol{u}^2 & 0 \\ 0 & -\boldsymbol{u}\,\boldsymbol{v} & 0 & \boldsymbol{u}^2 \\ -\boldsymbol{v}^2 & 0 & \boldsymbol{u}\,\boldsymbol{v} & 0 \\ 0 & -\boldsymbol{v}^2 & 0 & \boldsymbol{u}\,\boldsymbol{v} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} .$$

Carrying out Gaussian elimination on this system of equations leads to a unique solution

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \tfrac{1}{(\boldsymbol{u}\,\boldsymbol{v})^2 - \boldsymbol{u}^2\boldsymbol{v}^2} \begin{bmatrix} -\boldsymbol{u}\,\boldsymbol{v} \\ \boldsymbol{u}^2 \\ -\boldsymbol{v}^2 \\ \boldsymbol{u}\,\boldsymbol{v} \end{bmatrix} .$$

Substituting the scalar coefficients a, b, c, and d into the Eq.(D.4), the solution to the integral can be written as

$$\boldsymbol{f}\,(t) = \left\{ \frac{-(\boldsymbol{u}\,\boldsymbol{v})\boldsymbol{u} + \boldsymbol{u}^2\boldsymbol{v} + t(-\boldsymbol{v}^2\boldsymbol{u} + (\boldsymbol{u}\,\boldsymbol{v})\boldsymbol{v})}{((\boldsymbol{u}\,\boldsymbol{v})^2 - \boldsymbol{u}^2\boldsymbol{v}^2)||\boldsymbol{u} + t\boldsymbol{v}||} \right\} \Bigg|_0^1 . \tag{D.7}$$

The resulting electric field at $\boldsymbol{x}'$, exerted by the line charge extending from $\boldsymbol{a}$ to $\boldsymbol{b}$, is then given by

$$\boldsymbol{E}_{ab} = \frac{k_c L}{(\boldsymbol{u}^2\boldsymbol{v}^2 - (\boldsymbol{u}\,\boldsymbol{v})^2)} \left\{ \frac{(\boldsymbol{u}\,\boldsymbol{v} + \boldsymbol{v}^2)\boldsymbol{u} - (\boldsymbol{u}^2 + \boldsymbol{u}\,\boldsymbol{v})\boldsymbol{v}}{||\boldsymbol{u} + \boldsymbol{v}||} - \frac{(\boldsymbol{u}\,\boldsymbol{v})\boldsymbol{u} - \boldsymbol{u}^2\boldsymbol{v}}{||\boldsymbol{u}||} \right\} \tag{D.8}$$

Noting that (see Figure 4.8)

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{x}' - \boldsymbol{a} \overset{\text{def}}{=} \boldsymbol{x}_a \\ \boldsymbol{u} + \boldsymbol{v} &= \boldsymbol{x}' - \boldsymbol{b} \overset{\text{def}}{=} \boldsymbol{x}_b \\ \boldsymbol{v} &= \boldsymbol{a} - \boldsymbol{b} \overset{\text{def}}{=} \boldsymbol{x}_b - \boldsymbol{x}_a, \end{aligned}$$

and substituting the above vectors into Eq.(D.8), we obtain

$$\begin{aligned}\boldsymbol{E}_{ab}(\boldsymbol{x}') \quad = \quad & \left\{ \frac{(\boldsymbol{x}_a \cdot (\boldsymbol{x}_b - \boldsymbol{x}_a) + ||\boldsymbol{x}_b - \boldsymbol{x}_a||^2)\boldsymbol{x}_a - (||\boldsymbol{x}_a||^2 + \boldsymbol{x}_a \cdot (\boldsymbol{x}_b - \boldsymbol{x}_a))(\boldsymbol{x}_b - \boldsymbol{x}_a)}{||\boldsymbol{x}_b||} \right. \\ & \left. - \frac{\boldsymbol{x}_a \cdot (\boldsymbol{x}_b - \boldsymbol{x}_a)\boldsymbol{x}_a - ||\boldsymbol{x}_a||^2(\boldsymbol{x}_b - \boldsymbol{x}_a)}{||\boldsymbol{x}_a||} \right\} \frac{k_c L}{||\boldsymbol{x}_a||^2||\boldsymbol{x}_b - \boldsymbol{x}_a||^2 - (\boldsymbol{x}_a \cdot (\boldsymbol{x}_b - \boldsymbol{x}_a))^2}, \end{aligned} \tag{D.9}$$

which can be simplified to Eq.(4.31) rewritten here for convenience

$$\boldsymbol{E}_{ab}(\boldsymbol{x}') = \frac{k_c\, L}{||\boldsymbol{x}_a||^2||\boldsymbol{x}_b||^2 - (\boldsymbol{x}_a \cdot \boldsymbol{x}_b)^2} \left(\frac{||\boldsymbol{x}_b||^2\boldsymbol{x}_a - (\boldsymbol{x}_a \cdot \boldsymbol{x}_b)\boldsymbol{x}_b}{||\boldsymbol{x}_b||} + \frac{||\boldsymbol{x}_a||^2\boldsymbol{x}_b - (\boldsymbol{x}_a \cdot \boldsymbol{x}_b)\boldsymbol{x}_a}{||\boldsymbol{x}_a||} \right). \tag{D.10}$$